NASA Contractor Report 191009

Army Research Laboratory
Contractor Report ARL–CR–121

Manual for Automatic Generation of Finite Element Models of Spiral Bevel Gears in Mesh

G.D. Bibel
University of North Dakota
Grand Forks, North Dakota

and

S. Reddy and A. Kumar
University of Akron
Akron, Ohio

April 1994

Prepared for
Lewis Research Center
Under Grant NAG3–1476

National Aeronautics and
Space Administration

U.S. ARMY
RESEARCH LABORATORY

Manual for Automatic Generation of Finite Element Models of Spiral Bevel Gears in Mesh

G.D. Bibel
University of North Dakota
Grand Forks, North Dakota

and

S. Reddy and A. Kumar
University of Akron
Akron, Ohio

ABSTRACT

The goal of this research is to develop computer programs that generate finite element models suitable for doing 3D contact analysis of faced milled spiral bevel gears in mesh. A pinion tooth and a gear tooth are created and put in mesh.

There are two programs:

1. Points.f
2. Pat.f

Points.f is based on the equation of meshing for spiral bevel gears. It uses machine tool settings to solve for an N x M mesh of points on the four surfaces; pinion concave and convex, and gear concave and convex. Points.f creates the file POINTS.OUT, an ASCI file containing N x M points for each surface. (N is the number of node points along the length of the tooth, and M is nodes on the height.)

Pat.f reads POINTS.OUT and creates the file t1.out. T1.out is a series of PATRAN input commands. In addition to the mesh density on the tooth face, additional user specified variables are the number of finite elements through the thickness, and the number of finite elements along the tooth full fillet. A full fillet is assumed to exist for both the pinion and gear.

This report is based on the theory presented in Army Research Laboratory Report ARL-TR-158 "Contact Stress Analysis of Spiral Bevel Gears Using Nonlinear Finite Element Static Analysis" by G.D. Bibel, A. Kumar and S. Reddy; and AVSCOM Technical Report 91-C-020 "A Method for Determining Spiral-Bevel Gear Tooth Geometry for Finite Element Analysis" by R. F. Handschuh and F. L. Litvin.

TABLE OF CONTENTS

SUMMARY

The goal of this research is to develop computer programs that generate finite element models suitable for doing 3D contact analysis of faced milled spiral bevel gears in mesh. A pinion tooth and a gear tooth are created and put in mesh.

There are two programs:

1. Points.f
2. Pat.f

Points.f is based on the equation of meshing for spiral bevel gears. It uses machine tool settings to solve for an N x M mesh of points on the four surfaces; pinion concave and convex, and gear concave and convex. Points.f creates the file POINTS.OUT, an ASCI file containing N x M points for each surface. (N is the number of node points along the length of the tooth, and M is nodes on the height.)

NOTE: For Unix based systems, the program titles are case sensitive. All titles are lower case except POINTS.OUT.

Pat.f reads POINTS.OUT and creates the file t1.out. T1.out is a series of PATRAN input commands. In addition to the mesh density on the tooth face, additional user specified variables are the number of finite elements through the thickness, and the number of finite elements along the tooth full fillet. A full fillet is assumed to exist for both the pinion and gear.

This report is based on the theory presented in Army Research Laboratory Report ARL-TR-158 "Contact Stress Analysis of Spiral Bevel Gears Using Nonlinear Finite Element Static Analysis" by G.D. Bibel, A. Kumar and S. Reddy; and AVSCOM Technical Report 91-C-020 "A Method for Determining Spiral-Bevel Gear Tooth Geometry for Finite Element Analysis" by R. F. Handschuh and F. L. Litvin.

The following topics are covered in this report:

1. A description of the detailed procedure for generating a finite element model.

2. Instructions for inserting the data.

3. A numerical example of input and output.

4. A listing of the programs.

5. Sample plots.

Finite element analysis of spiral bevel gears can be used to determine contact stresses, bending stresses, stiffness for dynamic analysis, load sharing, contact area, and thermal gradients.

PROCEDURE

1. Prepare the input data for points.f (described elsewhere in this report) for the pinion.

2. Compile and execute points.f

3. During the execution of points.f, the user is prompted for the desired N x M mesh.

4. Execution of points.f creates the file POINTS.OUT. This file will contain N x M points on the pinion concave surface and N x M points on the pinion convex surface. (2 x N x M total points)

5. Prepare the input data for model.f (described elsewhere in this report) for the pinion.

6. Compile and execute model.f.

7. During the execution of model.f, the user is prompted for the desired N x M mesh. These values must be the same used in step 3 above. The user is also prompted for the number of finite elements through the tooth thickness (this must be an even number), and the number of finite elements along the length of 1/2 of the full fillet. Trial and error may be required to obtain finite elements with appropriate aspect ratios.

8. Execution of model.f creates t1.out. This file is suitable for direct input into PATRAN ver 2.5.

NOTE: An 8 x 6 mesh is suggested as a practice mesh. Accuracy is affected with coarser meshes. (The increment is too large for the numerical solution)

NUMERICAL EXAMPLE (SUMMARY)

A 8 x 6 mesh with 4 elements through the tooth thickness and 4 elements in the fillet region will be used as an example.

1. The input data is as shown in the section INPUT DATA: POINTS.F

2. Attachment 1 shows the output from points.f (i. e. the file POINTS.OUT). This file is the X, Y, and Z coordinates of the N x M mesh of points on the four surfaces.

3. The input data from the next step is as shown in the section INPUT DATA: PAT.F

4. Attachment 2 shows the output from Pat.f (i. e. the file t1.out.

5. Attachment 3 shows a typical PATRAN plot after reading t1.out.

INPUT DATA: Points.f

The input data for Points.f is as shown below. (The input data occurs inside of points.f) Each input variable is self explanatory or explained in detail in the reference reports.

The initial guess is very sensitive. The solution technique may converge on another surface if the initial guess is too far from the correct solution.

```
C
C
C
C
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C                  DESCRIPTION OF INPUT DATA
C-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C
C                1) RADIUS OF CUTTER, INCHES           (R)
C                2) CRADLE ANGLE, DEGREES              (Q)
C                3) BLADE ANGLE, DEGREES               (PSI)
C                4) CRADLE TO CUTTER DISTANCE, INCHES  (S)
C                5) RATIO OF ROLL                      (MCW)
C                6) MACHINE OFFSET, INCHES             (EM)
C                7) VECTOR SUM, INCHES                 (LM)
C                8) DEDENDUM ANGLE, DEGREES            (DEDEN)
C                9) PITCH ANGLE, DEGREES               (MU)
C               10) ADDENDUM ANGLE, DEGREES            (ADDAN)
C               11) CLEARANCE, INCHES                  (CL)
C               12) MEAN CONE DISTANCE, INCHES         (RL)
C               13) FACE WIDTH, INCHES                 (FW)
c               14) INIITIAL GUESS FOR SURFACE COORDINATE U      ( U(1) )
C               15) INITIAL GUESS FOR SURFACE COORDINATE THETA  ( THETA(1) )
C               16) INITIAL GUESS FOR ANGLE OF CRADLE, DEGREES  ( PHIC(1)  )
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C           INSERT CONCAVE SIDE OF PINION DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C
             R      = 2.96562137806
             Q      = 63.9420304635 * PI/180.0
             PSI    = 161.954330248 * PI/180.0
             S      = 2.94780202969
             MCW    = 0.30838512709
             EM    = 0.154575896
             LM    = 0.0384999977874
             DEDEN = 1.56666666 * pi/180
             MU    = 18.4333333 * pi/180
             ADDAN = 3.8833334  * pi/180
             CL    = 0.03
             RL    = 3.191
             FW    = 1.0
             U(1)  = 9.59703
          THETA(1) = 126.83544 * PI/180.0
          PHIC(1)  = -0.85813 * PI/180.0
C
           ELSEIF (INT .EQ. 2) THEN
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C             INSERT CONVEX SIDE OF PINION DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
               R     = 3.071306157
               Q     = 53.9259945467 * PI/180.0
               PSI   = 24.337423854 * PI/180.0
               S     = 2.80104946
               MCW   = 0.3220428536
               EM    = -0.17426159493
               LM    = -0.0518138227
               DEDEN = 1.56666666 * pi/180
               MU    = 18.4333333 * pi/180
               ADDAN = 3.8833334  * pi/180
               CL    = 0.03
               RL    = 3.191
               FW    = 1.0
               U(1)  = 7.42534
            THETA(1) = 124.43689* PI/180.0
            PHIC(1)  = -11.38663*PI/180.0
C
           ELSEIF (INT .EQ. 3) THEN
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C             INSERT CONCAVE SIDE OF GEAR DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
             R     = 3.0325
             Q     = 59.2342023 * PI/180.0
             PSI   = 158.0 * PI/180.0
             S     = 2.85995004691
             MCW   = 0.9508646
             EM    = 0.0
             LM    = 0.0
             DEDEN = 3.8833333333 * pi/180
             MU    = 71.5666666     * pi/180
             ADDAN = 1.5666666     * pi/180
             CL    = 0.0366
             RL    = 3.191
             FW    = 1.0
             U(1)  = 8.12602
          THETA(1) = 233.98994 * PI/180.0
          PHIC(1)  = -0.35063 * PI/180.0
C-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C
C
C
C
           ELSE
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C             INSERT CONVEX SIDE OF GEAR DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
   R      = 2.9675
   Q      = 59.2342023 * PI/180.0
   PSI    = 22.0 * PI/180.0
   S      = 2.85995004691
   MCW    = 0.9508646
   EM     = 0.0
   LM     = 0.0
   DEDEN  = 3.8833333333 * pi/180
   MU     = 71.5666666    * pi/180
   ADDAN  = 1.5666666    * pi/180
   CL     = 0.0366
   RL     = 3.191
   FW     = 1.0
   U(1)   = 7.89156
THETA(1)  = 234.95451 * PI/180.0
PHIC(1)   = -12.3384 * PI/180.0
```

INPUT DATA: Pat. f.

The input data for Model.f is as shown below. Some of the data is redundant with Points.f The new variables are as follows:

1. "ROTCON". ROTCON is the rotation of each convex surface required to obtain the desired top land thickness.

2. "ROTINT". ROTINT is the rotation of the pinion required to eliminate interference with the gear.

3. "ROGEAR". ROGEAR is the rotation of the gear required to place the gear in mesh with the pinion. For the general case, the gear tooth is rotated 360/(Number of gear teeth) + 180 degrees CW about the Z axis (the gear's axis of rotation).

```
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C               DESCRIPTION OF INPUT DATA FOR PINION
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C         1. DEDENDUM, DEGREES                          (DEDEN)
C         2. PITCH ANGLE, DEGREES                       (MU)
C         3. ROTATION OF CONVEX SURFACE TO
C                    CREATE TOP LAND                    (ROTCON)
C         4. ROTATION OF PINION TO ELIMINATE
C                    INTERFERENCE                       (ROTINT)
C         5. PINION ID, INCHES                          (RI)
C         6. CLEARANCE, INCHES                          (CL)
C         7. NUMBER OF PINION TEETH                     (NTPIN)
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C            INPUT THE PINION DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
              DEDEN = 1.56666666 * PI/180.0
               MU   = 18.4333333 * PI/180.0
               ROTCON = 2.275
               ROTINT = -3.56
               RI   = 0.609375
               CL   = 0.03
               NTPIN = 12
C-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C
C
        ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN
C
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C       DESCRIPTION OF INPUT DATA FOR THE GEAR
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C        1. DEDENDUM, DEGREES                            (DEDEN)
C        2. PITCH ANGEL, DEGREES                         (MU)
C        3. ROTATION OF CONVEX SURFACE TO
C                CREATE TOP LAND                         (ROTCON)
C        4. ROTATION OF GEAR TO PUT IN MESH              (ROGEAR)
C        5. CLEARANCE, INCHES                            (CL)
C        6. ID OF GEAR BASE                              (R1)
C        7. OD OF GEAR BASE                              (R2)
C        8. NUMBER OF GEAR TEETH                         (NTGE)
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C           INPUT THE GEAR DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
             DEDEN = 3.883333333 * PI/180.0
              MU   = 71.5666666 * PI/180.0
              ROTCON = -8.49
              ROGEAR = 190.0
              CL = 0.0366
              R1 = 2.375
              R2 = 3.250
              NTGE = 36
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
```

OUTPUT FROM POINTS.F

```
         X                       Y                          Z

0.7948185893337920      0.1518771876505158         2.565573009266192
0.8343433613077650      0.1745810495998836         2.552471190313917
0.8720324096583479      0.2042199804427480         2.539369371362535
0.9077908578945182      0.2394490290397444         2.526267552535221
0.9414155876464360      0.2795805696115394         2.513165735668416
0.9726798641190831      0.3241375989552031         2.500063936000037
0.8472371339193625      7.610602922158893E-02      2.702284801203873
0.8913330776440134      9.583903087412704OE-02     2.688392059145086
0.9342070276559395      0.1231797235370555         2.674499317051919
0.9756155352323542      0.1566714188085960         2.660606575085474
 1.015268047213935      0.1955944000191652         2.646713837702351
 1.052870943839311      0.2394597732991732         2.632821147263364
0.8920544861387928     -8.797704145004470E-03      2.838996593145300
0.9405003839933272      7.308351971982762E-03      2.824312928070127
0.9884537411081434      3.163027692243947E-02      2.809629262838540
 1.035498881976111      6.262057964140943E-02      2.794945597192247
 1.081249167998568      9.953939752452578E-02      2.780261940615036
 1.125338129677023      0.1418941391352297         2.765578408798671
0.9279228340072788     -0.1023212288383641         2.975708385187462
0.9804198623141954     -9.048815198583959E-02      2.960233797221159
 1.033263802256655     -6.991428008202849E-02      2.944759208992080
 1.085845886896256     -4.221124233563466E-02      2.929284618267694
 1.137675976466795     -8.125221209773325E-03      2.913810043739986
 1.188306404191815      3.185542469737212E-02      2.898335812878913
0.9534599334795802     -0.2037951651625605         3.112420178263912
 1.009630926058035     -0.1968645393407975         3.096154666768726
 1.067089798650322     -0.1807700720952332         3.079889156026669
 1.125017872008955     -0.1571554097480132         3.063623637059750
 1.182815359830648     -0.1267574497917443         3.047358144908540
 1.239934504589937     -9.005461852766361E-02      3.031093686081570
0.9672556523581568     -0.3123758786108948         3.249131977522822
 1.026644599586983     -0.3109539641202601         3.232075536925217
 1.088352545835276     -0.3000651176158047         3.215019104897882
 1.151339425325948     -0.2813479710083238         3.197962650909841
 1.214890364286590     -0.2555128377780433         3.180906240112322
 1.278297192284771     -0.2230290530267571         3.163853372103641
0.9678796662918135     -0.4270252087603694         3.385843803228283
 1.029951520163903     -0.4316883253646147         3.367996407956596
 1.095449239626837     -0.4267184461595268         3.350149057378801
 1.163106632451191     -0.4137081339625599         3.332301654167603
 1.232086458109564     -0.3933218937462701         3.314454332645862
 1.301293568935084     -0.3660335888044650         3.296621207705554
0.9538905193238456     -0.5464866609459378         3.522555720376076
 1.018031477753651     -0.5577745718372567         3.503917280198983
 1.086763175986946     -0.5594158588267257         3.485279016743563
 1.158596913336253     -0.5529133643964257         3.466640634659134
 1.232554428724539     -0.5388625992810234         3.448002511532153
 1.306217203742137     -0.5177479993723693         3.429430372498238
0.7093463178422439      0.3893983454546772         2.565573009266191
0.7570852600711056      0.3917005187907718         2.552471190313766
0.8073029755824992      0.3878249981807325         2.539369371359211
0.8593909679113807      0.3779778338261695         2.526267552392368
0.9128138134481132      0.3622148457047225         2.513165733397398
0.9670614194895887      0.3405352495955354         2.500063914362740
0.7805547654008799      0.3381377633245926         2.702284801203850
0.8313192150683369      0.3355117554273983         2.688392059127966
0.8840811061706690      0.3260622820926806         2.674499317054876
0.9382129090097685      0.3100453874934890         2.660606574980783
0.9931445219848133      0.2875595701230784         2.646713832903523
 1.048324166687691      0.2586450939788132         2.632821090805444
0.8483132970638206      0.2760491911215923         2.838996593141508
0.9015956233205106      0.2678053743188284         2.824312927885464
0.9562796868411379      0.2521317789571000         2.809629262692835
 1.011723722807169      0.2293346362776858         2.794945597542682
 1.067325789653079      0.1995595783460700         2.780261932355890
 1.122494542284766      0.1628943693243237         2.765578267158041
```

```
0.9112000448284689       0.2030391581389535      2.975708385079163
0.9663724749316263       0.1885085186004414      2.960233796271262
 1.022236225128360       0.1659976000986125      2.944759207797051
 1.078139321839011       0.1358626773835785      2.929284619894748
 1.133450935693196       9.8300989821880069E-02  2.913810031777249
 1.187542779161388       5.3454488313127157E-02  2.898335443486139
0.9676895991379796       0.1191435063835695      3.112420177016817
 1.023995422930553       9.7691650985565203E-02  3.096154663323241
 1.080168552957352       6.7782050056511611E-02  3.079889150748298
 1.135549961729417       2.9820889650890691E-02  3.063623641171534
 1.189482972857256      -1.5938341746537521E-02  3.047358130942706
 1.241305512978413      -6.9292504478944305E-02  3.031092619792399
 1.016149494629492       2.4546941529382415E-02  3.249131968954469
 1.072693870906420      -4.4094230428219916E-03  3.232075526708658
 1.128170570501692      -4.2209828439173868E-02  3.215019087512731
 1.181915774012974      -8.8397895323604203E-02  3.197962659109970
 1.233250099226129      -0.1426588841462646      3.180906229149712
 1.281481427875818      -0.2047210348378159      3.163849795972270
 1.054836149594651      -8.0393756502517454E-02  3.385843760892120
 1.110576450723173      -0.1173673033750138      3.367996381370604
 1.164207315546032      -0.1634608596042426      3.350149009219072
 1.215061572522443      -0.2181667004477261      3.332301668710605
 1.262440437691158      -0.2811032336663852      3.314454324809642
 1.305626160434559      -0.3519220612269351      3.296606971745099
 1.081890093168536      -0.1951217417487592      3.522555552829777
 1.135625698876967      -0.2405311921112059      3.503917217031341
 1.186109140444834      -0.2952059400453120      3.485278897558544
 1.232670689693987      -0.3585843827438959      3.466640659723548
 1.274595758431425      -0.4302115809299476      3.448002414715995
 1.311146184992092      -0.5096536842811183      3.429364146542768
 2.474433967420987      -0.3395653327372070     0.9856358798469023
 2.487839996829133      -0.3593061900500794     0.9465879479135923
 2.500980732115327      -0.3799719362136913     0.9075400160259967
 2.513847962185435      -0.4014972153625989     0.8684920851879474
 2.526433333215689      -0.4238257450684864     0.8294441631803695
 2.538728501263516      -0.4469082097766504     0.7903962834276070
 2.615744367457404      -0.2713527297527838      1.039882343666109
 2.630668874615893      -0.2921776971646133     0.9983126342013517
 2.645336856855221      -0.3140021877370152     0.9567429270473953
 2.659736900244837      -0.3367554455147062     0.9151732405016411
 2.673857752554951      -0.3603767778219593     0.8736036480711737
 2.687688379068866      -0.3848131384418187     0.8320343503411255
 2.755122082966633      -0.1939252488334882      1.094128807487018
 2.771646356120490      -0.2158264544757134      1.050037320781535
 2.787926437149529      -0.2388005015425003      1.005945850025553
 2.803947251406170      -0.2627715238015491     0.9618544915078790
 2.819694200468521      -0.2876746263983043     0.9177635422435383
 2.835153088849297      -0.3134531506865259     0.8736736400533243
 2.892115805626759      -0.1070309492253978      1.148375271344924
 2.910326750723583      -0.1300043557812005      1.101762007915943
 2.928309186669705      -0.1541221288023844      1.055148803591732
 2.946043871332245      -0.1793036410217184      1.008535978909207
 2.963512305696952      -0.2054800085262900     0.9619243699325180
 2.980696572015054      -0.2325910821084665     0.9153154443440815
 3.026234802111897      -1.0397603116659048E-02  1.202621735548316
 3.046226387571472      -3.4443711328922432E-02  1.153486695724431
 3.066008199637873      -5.9703409974563026E-02  1.104351817871708
 3.085556228980730      -8.6091707633022540E-02  1.055217942501711
 3.104847356582463      -0.1135359377283138      1.006086988192335
 3.123859380461631      -0.1419726571295392     0.9569616304362697
 3.156942158964726       9.6270882267980528E-02  1.256868201705318
 3.178817192398615       7.1146146344736038E-02  1.205211384146218
 3.200503834343097       4.4741446463306910E-02  1.153554939872925
 3.221972538278004       1.7145708246413704E-02  1.101900776270329
 3.243194610216827      -1.1564815230486136E-02  1.050252743854398
 3.264143329575414      -4.1323831199262528E-02 0.9986146558233269
```

```
3.283646305240354     0.2132989412203852      1.311114675992770
3.307518658058060     0.1870830974730224      1.256936072960456
3.331226092948566     0.1595243725147912      1.202758242897250
3.354732453389843     0.1307151794108270      1.148585121565755
3.378002211439905     0.1007351583096741      1.094423714581283
3.401005110789191     6.9652404241723697E-02  1.040277353579933
3.405690273410178     0.3410450079877778      1.365361177977477
3.431687717020426     0.3137175697464407      1.308660761835998
3.457545044455680     0.2849884724816092      1.251961842372733
3.483217889568247     0.2549531672766858      1.195272024233893
3.508662303297751     0.2236944341848668      1.138603073146720
3.533848840376255     0.1912798603623096      1.081952125201376
2.487328996883535    -0.2265447761173172     0.9856358798469033
2.504812553042866    -0.2106249324068905     0.9465879479128150
2.522236712927098    -0.1939207541553278     0.9075400159307193
2.539588928875214    -0.1764076751718493     0.8684920834049786
2.556855251703354    -0.1580661669201349     0.8294441489352412
2.574020473993094    -0.1388819032188771     0.7903962112604109
2.625150758261012    -0.1559948596305243      1.039882343666084
2.643236496908251    -0.1381569984025237     0.9983126341387127
2.661226646330261    -0.1194872621944678     0.9567429246234905
2.679106798755714    -9.9957306928012634E-02 0.9151732151109059
2.696861137105945    -7.9543963548449170E-02 0.8736035055753604
2.714472579493550    -5.8229565504748226E-02 0.8320337957402701
2.760887257218101    -7.6198734391658362E-02  1.094128807485250
2.779466509414771    -5.6307253677417623E-02  1.050037319991277
2.797909320045199    -3.5538740415832492E-02  1.005945831744241
2.816199124496761    -1.3861014156704154E-02 0.9618543448409637
2.834317939922617     8.7529198603615299E-03 0.9177628586525584
2.852246515854574     3.2324360398793938E-02 0.8736713719399361
2.894065956053995     1.3103412573647332E-02  1.148375271304306
2.913016385829245     3.5191943129389446E-02  1.101762003463058
2.931784666433014     5.8200237162429946E-02  1.055148723338887
2.950351750461571     8.2164289773184240E-02  1.008535449229927
2.968697149878801     0.1071150365619262     0.9619221908432191
2.986799102129386     0.1330776555556383     0.9153089376521233
3.024172263795945     0.1121931743753137      1.202621735122491
3.043355635337684     0.1366311709230543      1.153486678790011
3.062306015845269     0.1620292643340986      1.104351548799287
3.081001521818567     0.1884271541347537      1.055216417563395
3.099418775000900     0.2158597377906348      1.006081371459733
3.117533104008592     0.2443562555318990     0.9569463901021713
3.150641786577461     0.2213780095229714      1.256868198935295
3.169901202815439     0.2483283103564045      1.205211332798929
3.188871257383812     0.2762767736890748      1.153554174173141
3.207526851434866     0.3052666000315338      1.101896921856651
3.225841277669069     0.3353365347774662      1.050239962537327
3.243786499313236     0.3665199653141347     0.9985833243463247
3.272850598337732     0.3409963653901853      1.311114662721890
3.292007036268992     0.3706340385418412      1.256935938473254
3.310811656226333     0.4013059488337771      1.202756289922346
3.329235738826148     0.4330585061056649      1.148576146108771
3.347248740903902     0.4659339811534609      1.094396800991818
3.364818733544274     0.4999698305087152      1.040218589172536
3.390102864455341     0.4714241905482122      1.365361126402474
3.408950759800238     0.5039388434524499      1.308660443733778
3.427377505405820     0.5375223179436728      1.251957231553233
3.445350326286120     0.5722238817764125      1.195252233992485
3.462834246922116     0.6080886634937985      1.138549123343161
3.479792835690703     0.6451578176214294      1.081849345122198
```

Output from Pat. f.

```
 SET,LABEL,OFF
 VI
 1
 120,0,120
GRID,   1,,  0.802715/  0.102231/  2.565573
GRID,   2,,  0.843574/  0.122437/  2.552471
GRID,   3,,  0.883030/  0.149678/  2.539369
GRID,   4,,  0.920907/  0.182619/  2.526268
GRID,   5,,  0.956959/  0.220585/  2.513166
GRID,   6,,  0.990930/  0.263115/  2.500064
GRID,   7,,  0.850328/  0.023351/  2.702285
GRID,   8,,  0.895564/  0.040308/  2.688392
GRID,   9,,  0.940053/  0.064934/  2.674499
GRID,  10,,  0.983461/  0.095789/  2.660607
GRID,  11,,  1.025454/  0.132175/  2.646714
GRID,  12,,  1.065708/  0.173621/  2.632821
GRID,  13,,  0.889787/ -0.064172/  2.838997
GRID,  14,,  0.939139/ -0.051105/  2.824313
GRID,  15,,  0.988510/ -0.029808/  2.809629
GRID,  16,,  1.037389/ -0.001798/  2.794946
GRID,  17,,  1.085343/  0.032209/  2.780262
GRID,  18,,  1.131977/  0.071744/  2.765578
GRID,  19,,  0.919779/ -0.159742/  2.975708
GRID,  20,,  0.972909/ -0.151191/  2.960234
GRID,  21,,  1.026929/ -0.133939/  2.944759
GRID,  22,,  1.081129/ -0.109554/  2.929285
GRID,  23,,  1.134976/ -0.078752/  2.913810
GRID,  24,,  1.187991/ -0.041992/  2.898336
GRID,  25,,  0.938966/ -0.262606/  3.112420
GRID,  26,,  0.995459/ -0.259176/  3.096155
GRID,  27,,  1.053806/ -0.246681/  3.079889
GRID,  28,,  1.113089/ -0.226709/  3.063624
GRID,  29,,  1.172662/ -0.199958/  3.047358
GRID,  30,,  1.231950/ -0.166873/  3.031094
GRID,  31,,  0.945993/ -0.371834/  3.249132
GRID,  32,,  1.005355/ -0.374102/  3.232076
GRID,  33,,  1.067620/ -0.367066/  3.215019
GRID,  34,,  1.131648/ -0.352296/  3.197963
GRID,  35,,  1.196680/ -0.330457/  3.180906
GRID,  36,,  1.261982/ -0.301973/  3.163853
GRID,  37,,  0.939496/ -0.486300/  3.385844
GRID,  38,,  1.001159/ -0.494809/  3.367996
GRID,  39,,  1.066839/ -0.493916/  3.350149
GRID,  40,,  1.135174/ -0.485131/  3.332302
GRID,  41,,  1.205286/ -0.469068/  3.314454
GRID,  42,,  1.276054/ -0.446129/  3.296621
GRID,  43,,  0.918116/ -0.604663/  3.522556
GRID,  44,,  0.981433/ -0.619912/  3.503917
GRID,  45,,  1.049930/ -0.625818/  3.485279
GRID,  46,,  1.122029/ -0.623788/  3.466641
GRID,  47,,  1.196716/ -0.614357/  3.448003
GRID,  48,,  1.271548/ -0.597857/  3.429430
GRID,  49,,  0.717900/  0.373393/  2.565573
GRID,  50,,  0.765679/  0.374624/  2.552471
GRID,  51,,  0.815797/  0.369623/  2.539369
GRID,  52,,  0.867651/  0.358610/  2.526268
GRID,  53,,  0.920707/  0.341653/  2.513166
GRID,  54,,  0.974455/  0.318763/  2.500064
GRID,  55,,  0.787941/  0.320548/  2.702285
GRID,  56,,  0.838634/  0.316785/  2.688392
GRID,  57,,  0.891171/  0.306154/  2.674499
GRID,  58,,  0.944930/  0.288927/  2.660607
GRID,  59,,  0.999343/  0.265215/  2.646714
GRID,  60,,  1.053861/  0.235071/  2.632821
GRID,  61,,  0.854291/  0.256956/  2.838997
GRID,  62,,  0.907375/  0.247519/  2.824313
```

```
GRID,  63,,  0.961693/  0.230623/  2.809629
GRID,  64,,  1.016612/  0.206588/  2.794946
GRID,  65,,  1.071533/  0.175574/  2.780262
GRID,  66,,  1.125865/  0.137681/  2.765578
GRID,  67,,  0.915524/  0.182554/  2.975708
GRID,  68,,  0.970357/  0.166790/  2.960234
GRID,  69,,  1.025702/  0.143032/  2.944759
GRID,  70,,  1.080915/  0.111651/  2.929285
GRID,  71,,  1.135370/  0.072858/  2.913810
GRID,  72,,  1.188443/  0.026810/  2.898335
GRID,  73,,  0.970118/  0.097413/  3.112420
GRID,  74,,  1.025929/  0.074703/  3.096155
GRID,  75,,  1.081417/  0.043542/  3.079889
GRID,  76,,  1.135933/  0.004348/  3.063624
GRID,  77,,  1.188826/ -0.042609/  3.047358
GRID,  78,,  1.239439/ -0.097112/  3.031093
GRID,  79,,  1.016444/  0.001753/  3.249132
GRID,  80,,  1.072325/ -0.028464/  3.232076
GRID,  81,,  1.126940/ -0.067499/  3.215019
GRID,  82,,  1.179636/ -0.114881/  3.197963
GRID,  83,,  1.229741/ -0.170279/  3.180906
GRID,  84,,  1.276568/ -0.233408/  3.163850
GRID,  85,,  1.052768/ -0.104029/  3.385844
GRID,  86,,  1.107665/ -0.142243/  3.367996
GRID,  87,,  1.160249/ -0.189528/  3.350149
GRID,  88,,  1.209863/ -0.245360/  3.332302
GRID,  89,,  1.255819/ -0.309344/  3.314454
GRID,  90,,  1.297406/ -0.381113/  3.296607
GRID,  91,,  1.077242/ -0.219335/  3.522556
GRID,  92,,  1.129946/ -0.265938/  3.503917
GRID,  93,,  1.179191/ -0.321731/  3.485279
GRID,  94,,  1.224319/ -0.386138/  3.466641
GRID,  95,,  1.264627/ -0.458687/  3.448002
GRID,  96,,  1.299387/ -0.538929/  3.429364
GRID,  97,, -2.495807/ -0.095275/  0.985636
GRID,  98,, -2.512437/ -0.078162/  0.946588
GRID,  99,, -2.528967/ -0.060092/  0.907540
GRID, 100,, -2.545376/ -0.041128/  0.868492
GRID, 101,, -2.561648/ -0.021324/  0.829444
GRID, 102,, -2.577764/ -0.000727/  0.790396
GRID, 103,, -2.623125/ -0.186989/  1.039882
GRID, 104,, -2.641439/ -0.169072/  0.998313
GRID, 105,, -2.659674/ -0.150126/  0.956743
GRID, 106,, -2.677806/ -0.130219/  0.915173
GRID, 107,, -2.695815/ -0.109409/  0.873604
GRID, 108,, -2.713678/ -0.087745/  0.832034
GRID, 109,, -2.746940/ -0.287443/  1.094129
GRID, 110,, -2.767017/ -0.268744/  1.050037
GRID, 111,, -2.787039/ -0.248946/  1.005946
GRID, 112,, -2.806979/ -0.228121/  0.961854
GRID, 113,, -2.826811/ -0.206331/  0.917764
GRID, 114,, -2.846511/ -0.183628/  0.873674
GRID, 115,, -2.866764/ -0.396806/  1.148375
GRID, 116,, -2.888687/ -0.377344/  1.101762
GRID, 117,, -2.910585/ -0.356715/  1.055149
GRID, 118,, -2.932423/ -0.334996/  1.008536
GRID, 119,, -2.954171/ -0.312250/  0.961924
GRID, 120,, -2.975802/ -0.288535/  0.915315
GRID, 121,, -2.982065/ -0.515261/  1.202622
GRID, 122,, -3.005928/ -0.495052/  1.153487
GRID, 123,, -3.029796/ -0.473611/  1.104352
GRID, 124,, -3.053629/ -0.451018/  1.055218
GRID, 125,, -3.077393/ -0.427340/  1.006087
GRID, 126,, -3.101054/ -0.402637/  0.956962
GRID, 127,, -3.092264/ -0.643006/  1.256868
GRID, 128,, -3.118169/ -0.622061/  1.205211
```

```
GRID, 129,, -3.144112/ -0.599824/  1.153555
GRID, 130,, -3.170046/ -0.576375/  1.101901
GRID, 131,, -3.195931/ -0.551786/  1.050253
GRID, 132,, -3.221729/ -0.526117/  0.998615
GRID, 133,, -3.196721/ -0.780258/  1.311115
GRID, 134,, -3.224783/ -0.758586/  1.256936
GRID, 135,, -3.252916/ -0.735562/  1.202758
GRID, 136,, -3.281068/ -0.711273/  1.148585
GRID, 137,, -3.309190/ -0.685789/  1.094424
GRID, 138,, -3.337241/ -0.659173/  1.040277
GRID, 139,, -3.294728/ -0.927256/  1.365361
GRID, 140,, -3.325076/ -0.904858/  1.308661
GRID, 141,, -3.355529/ -0.881056/  1.251962
GRID, 142,, -3.386028/ -0.855935/  1.195272
GRID, 143,, -3.416514/ -0.829569/  1.138603
GRID, 144,, -3.446946/ -0.802021/  1.081952
GRID, 145,, -2.492435/  0.160921/  0.985636
GRID, 146,, -2.509493/  0.144546/  0.946588
GRID, 147,, -2.526471/  0.127389/  0.907540
GRID, 148,, -2.543356/  0.109424/  0.868492
GRID, 149,, -2.560133/  0.090634/  0.829444
GRID, 150,, -2.576786/  0.071004/  0.790396
GRID, 151,, -2.628350/  0.086764/  1.039882
GRID, 152,, -2.645959/  0.068456/  0.998313
GRID, 153,, -2.663451/  0.049318/  0.956743
GRID, 154,, -2.680810/  0.029324/  0.915173
GRID, 155,, -2.698021/  0.008450/  0.873604
GRID, 156,, -2.715064/ -0.013321/  0.832034
GRID, 157,, -2.761936/  0.003419/  1.094129
GRID, 158,, -2.779985/ -0.016955/  1.050037
GRID, 159,, -2.797874/ -0.038203/  1.005946
GRID, 160,, -2.815586/ -0.060355/  0.961854
GRID, 161,, -2.833103/ -0.083438/  0.917763
GRID, 162,, -2.850404/ -0.107474/  0.873671
GRID, 163,, -2.892716/ -0.089362/  1.148375
GRID, 164,, -2.911077/ -0.111942/  1.101762
GRID, 165,, -2.929233/ -0.135437/  1.055149
GRID, 166,, -2.947162/ -0.159882/  1.008535
GRID, 167,, -2.964844/ -0.185307/  0.961922
GRID, 168,, -2.982255/ -0.211738/  0.915309
GRID, 169,, -3.020166/ -0.191846/  1.202622
GRID, 170,, -3.038698/ -0.216781/  1.153487
GRID, 171,, -3.056973/ -0.242669/  1.104352
GRID, 172,, -3.074966/ -0.269551/  1.055216
GRID, 173,, -3.092654/ -0.297459/  1.006081
GRID, 174,, -3.110011/ -0.326423/  0.956946
GRID, 175,, -3.143714/ -0.304325/  1.256868
GRID, 176,, -3.162257/ -0.331774/  1.205211
GRID, 177,, -3.180484/ -0.360212/  1.153554
GRID, 178,, -3.198369/ -0.389684/  1.101897
GRID, 179,, -3.215884/ -0.420226/  1.050240
GRID, 180,, -3.233002/ -0.451871/  0.998583
GRID, 181,, -3.262728/ -0.427123/  1.311115
GRID, 182,, -3.281097/ -0.457255/  1.256936
GRID, 183,, -3.299087/ -0.488411/  1.202756
GRID, 184,, -3.316668/ -0.520639/  1.148576
GRID, 185,, -3.333808/ -0.553977/  1.094397
GRID, 186,, -3.350475/ -0.588464/  1.040219
GRID, 187,, -3.376503/ -0.560595/  1.365361
GRID, 188,, -3.394487/ -0.593595/  1.308660
GRID, 189,, -3.412023/ -0.627652/  1.251957
GRID, 190,, -3.429075/ -0.662815/  1.195252
GRID, 191,, -3.445608/ -0.699128/  1.138549
GRID, 192,, -3.461584/ -0.736632/  1.081849
LINE,   1,ST,,   1,  49
LINE,  9/ 10,BR,.5,   1
```

```
LINE,   2,ST,,   7,  55
LINE, 11/ 12,BR,.5,   2
LINE,   3,ST,,  13,  61
LINE, 13/ 14,BR,.5,   3
LINE,   4,ST,,  19,  67
LINE, 15/ 16,BR,.5,   4
LINE,   5,ST,,  25,  73
LINE, 17/ 18,BR,.5,   5
LINE,   6,ST,,  31,  79
LINE, 19/ 20,BR,.5,   6
LINE,   7,ST,,  37,  85
LINE, 21/ 22,BR,.5,   7
LINE,   8,ST,,  43,  91
LINE, 23/ 24,BR,.5,   8
PATCH,    1, QUAD,,   1/   2/  50/  49
PATCH,    2, QUAD,,   2/   3/  51/  50
PATCH,    3, QUAD,,   3/   4/  52/  51
PATCH,    4, QUAD,,   4/   5/  53/  52
PATCH,    5, QUAD,,   5/   6/  54/  53
PATCH,    6, QUAD,,   7/   8/  56/  55
PATCH,    7, QUAD,,   8/   9/  57/  56
PATCH,    8, QUAD,,   9/  10/  58/  57
PATCH,    9, QUAD,,  10/  11/  59/  58
PATCH,   10, QUAD,,  11/  12/  60/  59
PATCH,   11, QUAD,,  13/  14/  62/  61
PATCH,   12, QUAD,,  14/  15/  63/  62
PATCH,   13, QUAD,,  15/  16/  64/  63
PATCH,   14, QUAD,,  16/  17/  65/  64
PATCH,   15, QUAD,,  17/  18/  66/  65
PATCH,   16, QUAD,,  19/  20/  68/  67
PATCH,   17, QUAD,,  20/  21/  69/  68
PATCH,   18, QUAD,,  21/  22/  70/  69
PATCH,   19, QUAD,,  22/  23/  71/  70
PATCH,   20, QUAD,,  23/  24/  72/  71
PATCH,   21, QUAD,,  25/  26/  74/  73
PATCH,   22, QUAD,,  26/  27/  75/  74
PATCH,   23, QUAD,,  27/  28/  76/  75
PATCH,   24, QUAD,,  28/  29/  77/  76
PATCH,   25, QUAD,,  29/  30/  78/  77
PATCH,   26, QUAD,,  31/  32/  80/  79
PATCH,   27, QUAD,,  32/  33/  81/  80
PATCH,   28, QUAD,,  33/  34/  82/  81
PATCH,   29, QUAD,,  34/  35/  83/  82
PATCH,   30, QUAD,,  35/  36/  84/  83
PATCH,   31, QUAD,,  37/  38/  86/  85
PATCH,   32, QUAD,,  38/  39/  87/  86
PATCH,   33, QUAD,,  39/  40/  88/  87
PATCH,   34, QUAD,,  40/  41/  89/  88
PATCH,   35, QUAD,,  41/  42/  90/  89
PATCH,   36, QUAD,,  43/  44/  92/  91
PATCH,   37, QUAD,,  44/  45/  93/  92
PATCH,   38, QUAD,,  45/  46/  94/  93
PATCH,   39, QUAD,,  46/  47/  95/  94
PATCH,   40, QUAD,,  47/  48/  96/  95
HPAT,     1,2P,,    1,   6
HPAT,     2,2P,,    2,   7
HPAT,     3,2P,,    3,   8
HPAT,     4,2P,,    4,   9
HPAT,     5,2P,,    5,  10
HPAT,     6,2P,,    6,  11
HPAT,     7,2P,,    7,  12
HPAT,     8,2P,,    8,  13
HPAT,     9,2P,,    9,  14
HPAT,    10,2P,,   10,  15
HPAT,    11,2P,,   11,  16
HPAT,    12,2P,,   12,  17
```

```
HPAT,  13,2P,,  13,  18
HPAT,  14,2P,,  14,  19
HPAT,  15,2P,,  15,  20
HPAT,  16,2P,,  16,  21
HPAT,  17,2P,,  17,  22
HPAT,  18,2P,,  18,  23
HPAT,  19,2P,,  19,  24
HPAT,  20,2P,,  20,  25
HPAT,  21,2P,,  21,  26
HPAT,  22,2P,,  22,  27
HPAT,  23,2P,,  23,  28
HPAT,  24,2P,,  24,  29
HPAT,  25,2P,,  25,  30
HPAT,  26,2P,,  26,  31
HPAT,  27,2P,,  27,  32
HPAT,  28,2P,,  28,  33
HPAT,  29,2P,,  29,  34
HPAT,  30,2P,,  30,  35
HPAT,  31,2P,,  31,  36
HPAT,  32,2P,,  32,  37
HPAT,  33,2P,,  33,  38
HPAT,  34,2P,,  34,  39
HPAT,  35,2P,,  35,  40
GRID, 201,,  0.604492/  0.076986/  2.565573
LINE, 25,ARC,5(0)/1/  20.22185, 201
GRID, 203,,  0.609145/  0.016728/  2.702285
LINE, 26,ARC,5(0)/1/  20.56433, 203
GRID, 205,,  0.607796/ -0.043834/  2.838997
LINE, 27,ARC,5(0)/1/  20.86543, 205
GRID, 207,,  0.600388/ -0.104272/  2.975708
LINE, 28,ARC,5(0)/1/  21.12931, 207
GRID, 209,,  0.586856/ -0.164129/  3.112420
LINE, 29,ARC,5(0)/1/  21.35904, 209
GRID, 211,,  0.567137/ -0.222920/  3.249132
LINE, 30,ARC,5(0)/1/  21.55671, 211
GRID, 213,,  0.541174/ -0.280122/  3.385844
LINE, 31,ARC,5(0)/1/  21.72357, 213
GRID, 215,,  0.508920/ -0.335170/  3.522556
LINE, 32,ARC,5(0)/1/  21.85998, 215
LINE,  33,ARC, 0.80851/ 0.03345/0/ 0.80851/ 0.03345/1.0/  170.22185,   1
LINE,  49/  50,BR,.5,  33
LINE,  34 ,ARC,0/0/0/0/0/1/  -9.778, 201
LINE,  81/  82,BR,.5,  34
LINE,  65,ST,, 193, 220
LINE,  97,ST,, 218, 220
LINE, 105,ST,,  49, 202
LINE, 113,ST,, 217, 219
LI, 121,MER,,  25/  81
LINE,  35,ARC, 0.84937/-0.04667/0/ 0.84937/-0.04667/1.0/  170.56433,   7
LINE,  51/  52,BR,.5,  35
LINE,  36 ,ARC,0/0/0/0/0/1/  -9.436, 203
LINE,  83/  84,BR,.5,  36
LINE,  66,ST,, 194, 224
LINE,  98,ST,, 222, 224
LINE, 106,ST,,  55, 204
LINE, 114,ST,, 221, 223
LI, 122,MER,,  26/  83
LINE,  37,ARC, 0.88185/-0.13482/0/ 0.88185/-0.13482/1.0/  170.86543,  13
LINE,  53/  54,BR,.5,  37
LINE,  38 ,ARC,0/0/0/0/0/1/  -9.135, 205
LINE,  85/  86,BR,.5,  38
LINE,  67,ST,, 195, 228
LINE,  99,ST,, 226, 228
LINE, 107,ST,,  61, 206
LINE, 115,ST,, 225, 227
LI, 123,MER,,  27/  85
```

```
LINE,  39,ARC, 0.90467/-0.23039/0/ 0.90467/-0.23039/1.0/  171.12931,  19
LINE,  55/  56,BR,.5,  39
LINE,  40 ,ARC,0/0/0/0/0/1/  -8.871, 207
LINE,  87/  88,BR,.5,  40
LINE,  68,ST,, 196, 232
LINE, 100,ST,, 230, 232
LINE, 108,ST,,  67, 208
LINE, 116,ST,, 229, 231
LI, 124,MER,,  28/  87
LINE,  41,ARC, 0.91651/-0.33260/0/ 0.91651/-0.33260/1.0/  171.35904,  25
LINE,  57/  58,BR,.5,  41
LINE,  42 ,ARC,0/0/0/0/0/1/  -8.641, 209
LINE,  89/  90,BR,.5,  42
LINE,  69,ST,, 197, 236
LINE, 101,ST,, 234, 236
LINE, 109,ST,,  73, 210
LINE, 117,ST,, 233, 235
LI, 125,MER,,  29/  89
LINE,  43,ARC, 0.91605/-0.44046/0/ 0.91605/-0.44046/1.0/  171.55671,  31
LINE,  59/  60,BR,.5,  43
LINE,  44 ,ARC,0/0/0/0/0/1/  -8.443, 211
LINE,  91/  92,BR,.5,  44
LINE,  70,ST,, 198, 240
LINE, 102,ST,, 238, 240
LINE, 110,ST,,  79, 212
LINE, 118,ST,, 237, 239
LI, 126,MER,,  30/  91
LINE,  45,ARC, 0.90195/-0.55283/0/ 0.90195/-0.55283/1.0/  171.72357,  37
LINE,  61/  62,BR,.5,  45
LINE,  46 ,ARC,0/0/0/0/0/1/  -8.276, 213
LINE,  93/  94,BR,.5,  46
LINE,  71,ST,, 199, 244
LINE, 103,ST,, 242, 244
LINE, 111,ST,,  85, 214
LINE, 119,ST,, 241, 243
LI, 127,MER,,  31/  93
LINE,  47,ARC, 0.87288/-0.66830/0/ 0.87288/-0.66830/1.0/  171.85998,  43
LINE,  63/  64,BR,.5,  47
LINE,  48 ,ARC,0/0/0/0/0/1/  -8.140, 215
LINE,  95/  96,BR,.5,  48
LINE,  72,ST,, 200, 248
LINE, 104,ST,, 246, 248
LINE, 112,ST,,  91, 216
LINE, 120,ST,, 245, 247
LI, 128,MER,,  32/  95
PA,  41,EDGE,,  10/ 105/ 121/  65
PA,  49,EDGE,,   9/  65/  97/  49
PA,  57,EDGE,,  82/ 113/  50/  97
PA,  42,EDGE,,  12/ 106/ 122/  66
PA,  50,EDGE,,  11/  66/  98/  51
PA,  58,EDGE,,  84/ 114/  52/  98
PA,  43,EDGE,,  14/ 107/ 123/  67
PA,  51,EDGE,,  13/  67/  99/  53
PA,  59,EDGE,,  86/ 115/  54/  99
PA,  44,EDGE,,  16/ 108/ 124/  68
PA,  52,EDGE,,  15/  68/ 100/  55
PA,  60,EDGE,,  88/ 116/  56/ 100
PA,  45,EDGE,,  18/ 109/ 125/  69
PA,  53,EDGE,,  17/  69/ 101/  57
PA,  61,EDGE,,  90/ 117/  58/ 101
PA,  46,EDGE,,  20/ 110/ 126/  70
PA,  54,EDGE,,  19/  70/ 102/  59
PA,  62,EDGE,,  92/ 118/  60/ 102
PA,  47,EDGE,,  22/ 111/ 127/  71
PA,  55,EDGE,,  21/  71/ 103/  61
PA,  63,EDGE,,  94/ 119/  62/ 103
```

```
PA,  48,EDGE,,  24/ 112/ 128/  72
PA,  56,EDGE,,  23/  72/ 104/  63
PA,  64,EDGE,,  96/ 120/  64/ 104
HPAT,  36 ,2P,,  41,  42
HPAT,  43 ,2P,,  49,  50
HPAT,  50 ,2P,,  57,  58
HPAT,  37 ,2P,,  42,  43
HPAT,  44 ,2P,,  50,  51
HPAT,  51 ,2P,,  58,  59
HPAT,  38 ,2P,,  43,  44
HPAT,  45 ,2P,,  51,  52
HPAT,  52 ,2P,,  59,  60
HPAT,  39 ,2P,,  44,  45
HPAT,  46 ,2P,,  52,  53
HPAT,  53 ,2P,,  60,  61
HPAT,  40 ,2P,,  45,  46
HPAT,  47 ,2P,,  53,  54
HPAT,  54 ,2P,,  61,  62
HPAT,  41 ,2P,,  46,  47
HPAT,  48 ,2P,,  54,  55
HPAT,  55 ,2P,,  62,  63
HPAT,  42 ,2P,,  47,  48
HPAT,  49 ,2P,,  55,  56
HPAT,  56 ,2P,,  63,  64
 SET,LINES,0
 NAME,PI 1
 SET,ACTIVE,NONE
LINE, 129,ST,,  97, 145
LINE,137/138,BR,.5, 129
LINE, 130,ST,, 103, 151
LINE,139/140,BR,.5, 130
LINE, 131,ST,, 109, 157
LINE,141/142,BR,.5, 131
LINE, 132,ST,, 115, 163
LINE,143/144,BR,.5, 132
LINE, 133,ST,, 121, 169
LINE,145/146,BR,.5, 133
LINE, 134,ST,, 127, 175
LINE,147/148,BR,.5, 134
LINE, 135,ST,, 133, 181
LINE,149/150,BR,.5, 135
LINE, 136,ST,, 139, 187
LINE,151/152,BR,.5, 136
PATCH,  65, QUAD,,  97/  98/ 146/ 145
PATCH,  66, QUAD,,  98/  99/ 147/ 146
PATCH,  67, QUAD,,  99/ 100/ 148/ 147
PATCH,  68, QUAD,, 100/ 101/ 149/ 148
PATCH,  69, QUAD,, 101/ 102/ 150/ 149
PATCH,  70, QUAD,, 103/ 104/ 152/ 151
PATCH,  71, QUAD,, 104/ 105/ 153/ 152
PATCH,  72, QUAD,, 105/ 106/ 154/ 153
PATCH,  73, QUAD,, 106/ 107/ 155/ 154
PATCH,  74, QUAD,, 107/ 108/ 156/ 155
PATCH,  75, QUAD,, 109/ 110/ 158/ 157
PATCH,  76, QUAD,, 110/ 111/ 159/ 158
PATCH,  77, QUAD,, 111/ 112/ 160/ 159
PATCH,  78, QUAD,, 112/ 113/ 161/ 160
PATCH,  79, QUAD,, 113/ 114/ 162/ 161
PATCH,  80, QUAD,, 115/ 116/ 164/ 163
PATCH,  81, QUAD,, 116/ 117/ 165/ 164
PATCH,  82, QUAD,, 117/ 118/ 166/ 165
PATCH,  83, QUAD,, 118/ 119/ 167/ 166
PATCH,  84, QUAD,, 119/ 120/ 168/ 167
PATCH,  85, QUAD,, 121/ 122/ 170/ 169
PATCH,  86, QUAD,, 122/ 123/ 171/ 170
PATCH,  87, QUAD,, 123/ 124/ 172/ 171
```

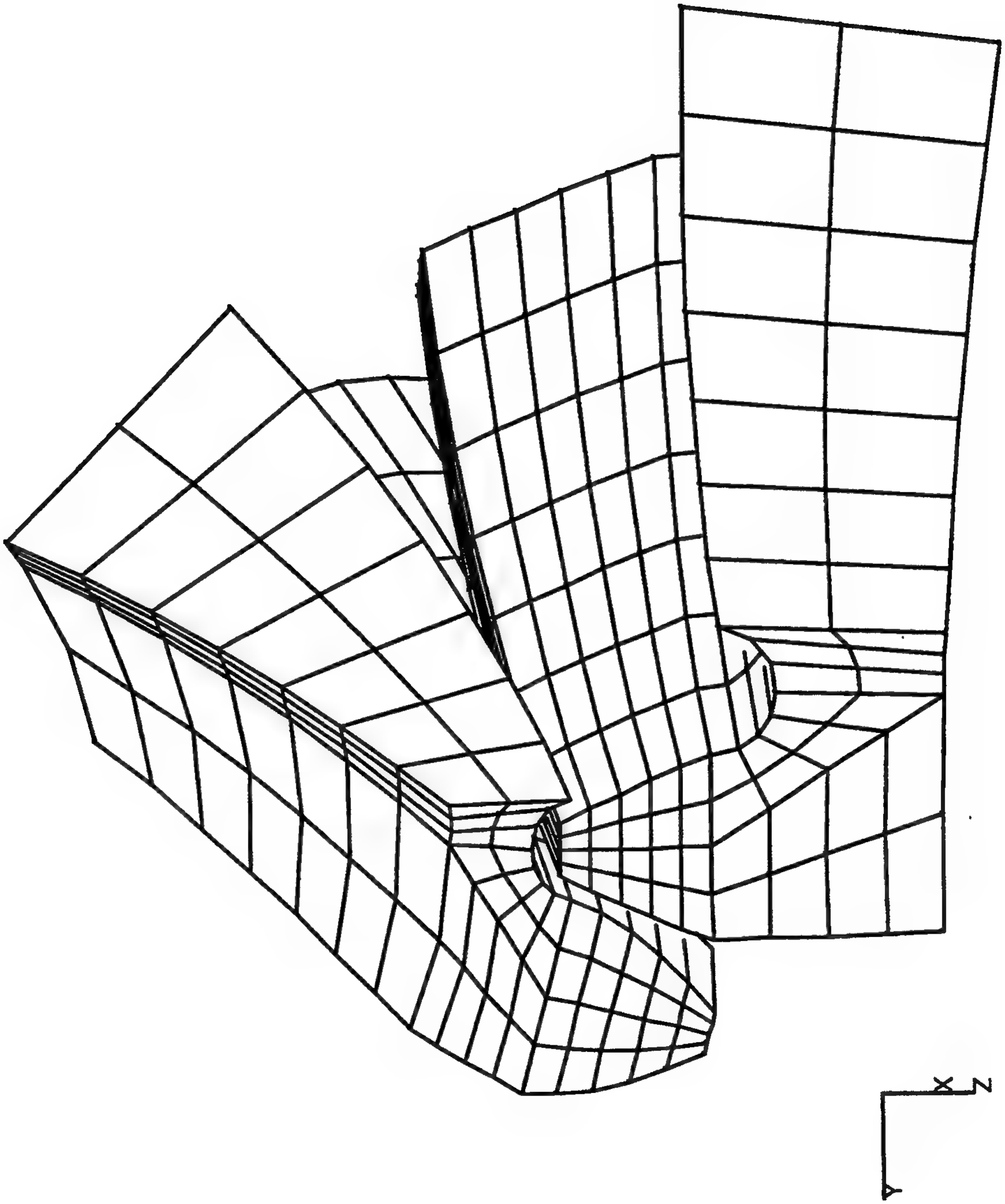
```
PATCH,  88, QUAD,, 124/ 125/ 173/ 172
PATCH,  89, QUAD,, 125/ 126/ 174/ 173
PATCH,  90, QUAD,, 127/ 128/ 176/ 175
PATCH,  91, QUAD,, 128/ 129/ 177/ 176
PATCH,  92, QUAD,, 129/ 130/ 178/ 177
PATCH,  93, QUAD,, 130/ 131/ 179/ 178
PATCH,  94, QUAD,, 131/ 132/ 180/ 179
PATCH,  95, QUAD,, 133/ 134/ 182/ 181
PATCH,  96, QUAD,, 134/ 135/ 183/ 182
PATCH,  97, QUAD,, 135/ 136/ 184/ 183
PATCH,  98, QUAD,, 136/ 137/ 185/ 184
PATCH,  99, QUAD,, 137/ 138/ 186/ 185
PATCH, 100, QUAD,, 139/ 140/ 188/ 187
PATCH, 101, QUAD,, 140/ 141/ 189/ 188
PATCH, 102, QUAD,, 141/ 142/ 190/ 189
PATCH, 103, QUAD,, 142/ 143/ 191/ 190
PATCH, 104, QUAD,, 143/ 144/ 192/ 191
HPAT,  57,2P,,  65,  70
HPAT,  58,2P,,  66,  71
HPAT,  59,2P,,  67,  72
HPAT,  60,2P,,  68,  73
HPAT,  61,2P,,  69,  74
HPAT,  62,2P,,  70,  75
HPAT,  63,2P,,  71,  76
HPAT,  64,2P,,  72,  77
HPAT,  65,2P,,  73,  78
HPAT,  66,2P,,  74,  79
HPAT,  67,2P,,  75,  80
HPAT,  68,2P,,  76,  81
HPAT,  69,2P,,  77,  82
HPAT,  70,2P,,  78,  83
HPAT,  71,2P,,  79,  84
HPAT,  72,2P,,  80,  85
HPAT,  73,2P,,  81,  86
HPAT,  74,2P,,  82,  87
HPAT,  75,2P,,  83,  88
HPAT,  76,2P,,  84,  89
HPAT,  77,2P,,  85,  90
HPAT,  78,2P,,  86,  91
HPAT,  79,2P,,  87,  92
HPAT,  80,2P,,  88,  93
HPAT,  81,2P,,  89,  94
HPAT,  82,2P,,  90,  95
HPAT,  83,2P,,  91,  96
HPAT,  84,2P,,  92,  97
HPAT,  85,2P,,  93,  98
HPAT,  86,2P,,  94,  99
HPAT,  87,2P,,  95, 100
HPAT,  88,2P,,  96, 101
HPAT,  89,2P,,  97, 102
HPAT,  90,2P,,  98, 103
HPAT,  91,2P,,  99, 104
GRID, 257,, -2.373271/ -0.090597/  1.284378
LINE, 153       ,ARC,5(0)/1.0/   -5.8802 , 257
GRID, 259,, -2.493672/ -0.177761/  1.356060
LINE, 154       ,ARC,5(0)/1.0/   -5.9681 , 259
GRID, 261,, -2.610745/ -0.273191/  1.427743
LINE, 155       ,ARC,5(0)/1.0/   -6.0447 , 261
GRID, 263,, -2.724029/ -0.377049/  1.499426
LINE, 156       ,ARC,5(0)/1.0/   -6.1112 , 263
GRID, 265,, -2.833021/ -0.489508/  1.571109
LINE, 157       ,ARC,5(0)/1.0/   -6.1685 , 265
GRID, 267,, -2.937172/ -0.610756/  1.642791
LINE, 158       ,ARC,5(0)/1.0/   -6.2174 , 267
GRID, 269,, -3.035876/ -0.740999/  1.714474
LINE, 159       ,ARC,5(0)/1.0/   -6.2584 , 269
```

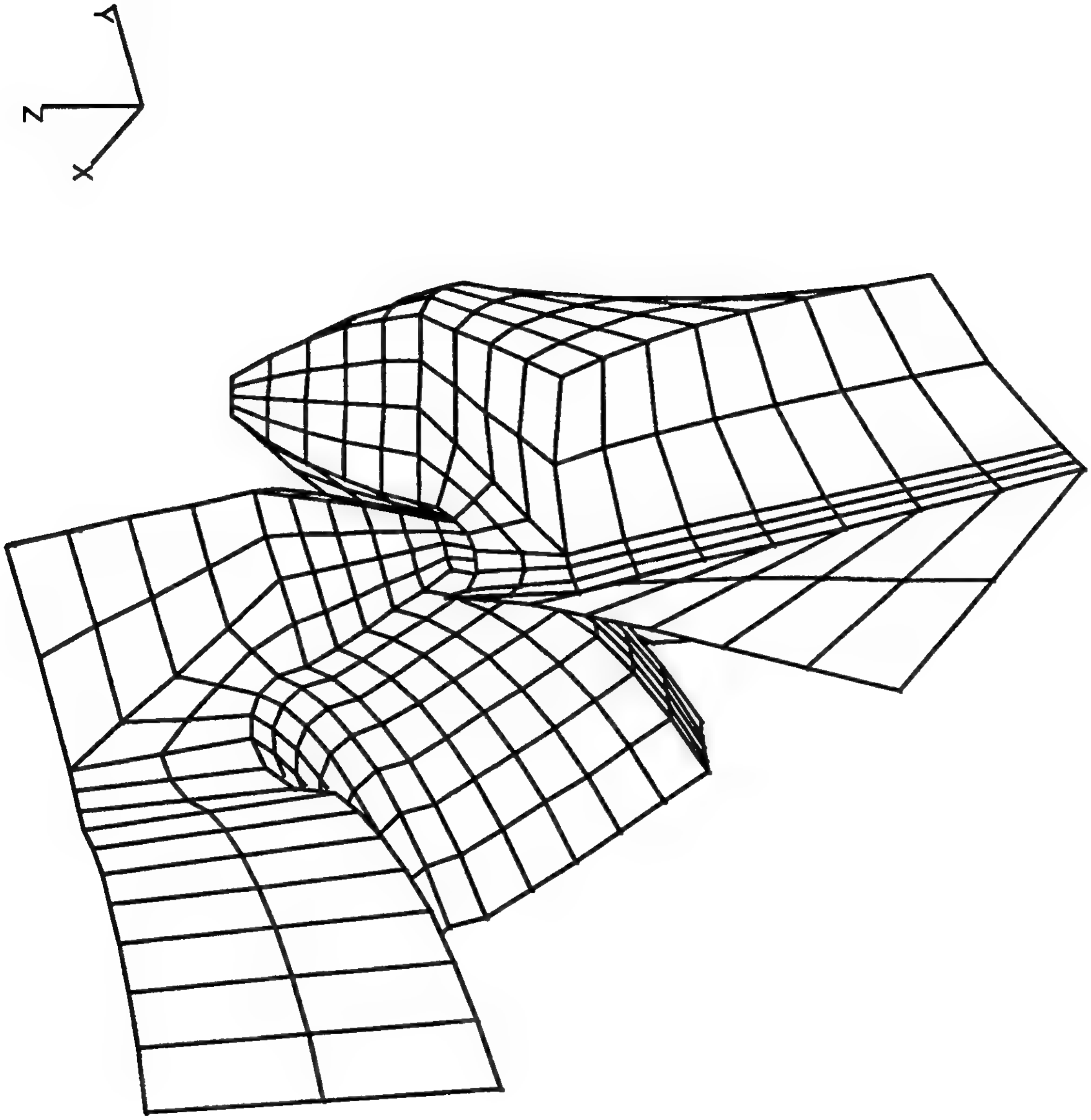
```
GRID, 271,, -3.128464/ -0.880463/  1.786157
LINE, 160       ,ARC,5(0)/1.0/   -6.2918 , 271
LINE, 161,ARC,0/0/0/-2.4908/-0.1849/  0.985636/-175.880,  97
LINE, 177/ 178,BR,.5, 161
LINE, 201 ,ARC,0/0/0/0/0/1/   4.120, 257
LINE, 218/ 219,BR,.5, 201
LINE, 193,ST,, 249, 276
LINE, 234,ST,, 274, 276
LINE, 242,ST,, 145, 258
LINE, 250,ST,, 273, 275
LI, 258,MER,, 153/ 218
LINE, 162,ARC,0/0/0/-2.6149/-0.2791/  1.039882/-175.968, 103
LINE, 179/ 180,BR,.5, 162
LINE, 203 ,ARC,0/0/0/0/0/1/   4.032, 259
LINE, 220/ 221,BR,.5, 203
LINE, 194,ST,, 250, 280
LINE, 235,ST,, 278, 280
LINE, 243,ST,, 151, 260
LINE, 251,ST,, 277, 279
LI, 259,MER,, 154/ 220
LINE, 163,ARC,0/0/0/-2.7354/-0.3821/  1.094129/-176.045, 109
LINE, 181/ 182,BR,.5, 163
LINE, 205 ,ARC,0/0/0/0/0/1/   3.955, 261
LINE, 222/ 223,BR,.5, 205
LINE, 195,ST,, 251, 284
LINE, 236,ST,, 282, 284
LINE, 244,ST,, 157, 262
LINE, 252,ST,, 281, 283
LI, 260,MER,, 155/ 222
LINE, 164,ARC,0/0/0/-2.8516/-0.4938/  1.148375/-176.111, 115
LINE, 183/ 184,BR,.5, 164
LINE, 207 ,ARC,0/0/0/0/0/1/   3.889, 263
LINE, 224/ 225,BR,.5, 207
LINE, 196,ST,, 252, 288
LINE, 237,ST,, 286, 288
LINE, 245,ST,, 163, 264
LINE, 253,ST,, 285, 287
LI, 261,MER,, 156/ 224
LINE, 165,ARC,0/0/0/-2.9632/-0.6147/  1.202622/-176.169, 121
LINE, 185/ 186,BR,.5, 165
LINE, 209 ,ARC,0/0/0/0/0/1/   3.831, 265
LINE, 226/ 227,BR,.5, 209
LINE, 197,ST,, 253, 292
LINE, 238,ST,, 290, 292
LINE, 246,ST,, 169, 266
LINE, 254,ST,, 289, 291
LI, 262,MER,, 157/ 226
LINE, 166,ARC,0/0/0/-3.0694/-0.7447/  1.256868/-176.217, 127
LINE, 187/ 188,BR,.5, 166
LINE, 211 ,ARC,0/0/0/0/0/1/   3.783, 267
LINE, 228/ 229,BR,.5, 211
LINE, 198,ST,, 254, 296
LINE, 239,ST,, 294, 296
LINE, 247,ST,, 175, 268
LINE, 255,ST,, 293, 295
LI, 263,MER,, 158/ 228
LINE, 167,ARC,0/0/0/-3.1695/-0.8842/  1.311115/-176.258, 133
LINE, 189/ 190,BR,.5, 167
LINE, 213 ,ARC,0/0/0/0/0/1/   3.742, 269
LINE, 230/ 231,BR,.5, 213
LINE, 199,ST,, 255, 300
LINE, 240,ST,, 298, 300
LINE, 248,ST,, 181, 270
LINE, 256,ST,, 297, 299
LI, 264,MER,, 159/ 230
LINE, 168,ARC,0/0/0/-3.2630/-1.0334/  1.365361/-176.292, 139
```

```
LINE, 191/ 192,BR,.5, 168
LINE, 215 ,ARC,0/0/0/0/0/1/   3.708, 271
LINE, 232/ 233,BR,.5, 215
LINE, 200,ST,, 256, 304
LINE, 241,ST,, 302, 304
LINE, 249,ST,, 187, 272
LINE, 257,ST,, 301, 303
LI, 265,MER,, 160/ 232
PA, 105,EDGE,, 242/ 138/ 193/ 258
PA, 113,EDGE,, 193/ 137/ 177/ 234
PA, 121,EDGE,, 234/ 178/ 250/ 219
PA, 106,EDGE,, 243/ 140/ 194/ 259
PA, 114,EDGE,, 194/ 139/ 179/ 235
PA, 122,EDGE,, 235/ 180/ 251/ 221
PA, 107,EDGE,, 244/ 142/ 195/ 260
PA, 115,EDGE,, 195/ 141/ 181/ 236
PA, 123,EDGE,, 236/ 182/ 252/ 223
PA, 108,EDGE,, 245/ 144/ 196/ 261
PA, 116,EDGE,, 196/ 143/ 183/ 237
PA, 124,EDGE,, 237/ 184/ 253/ 225
PA, 109,EDGE,, 246/ 146/ 197/ 262
PA, 117,EDGE,, 197/ 145/ 185/ 238
PA, 125,EDGE,, 238/ 186/ 254/ 227
PA, 110,EDGE,, 247/ 148/ 198/ 263
PA, 118,EDGE,, 198/ 147/ 187/ 239
PA, 126,EDGE,, 239/ 188/ 255/ 229
PA, 111,EDGE,, 248/ 150/ 199/ 264
PA, 119,EDGE,, 199/ 149/ 189/ 240
PA, 127,EDGE,, 240/ 190/ 256/ 231
PA, 112,EDGE,, 249/ 152/ 200/ 265
PA, 120,EDGE,, 200/ 151/ 191/ 241
PA, 128,EDGE,, 241/ 192/ 257/ 233
HPAT,  92 ,2P,, 105, 106
HPAT,  99 ,2P,, 113, 114
HPAT, 106 ,2P,, 121, 122
HPAT,  93 ,2P,, 106, 107
HPAT, 100 ,2P,, 114, 115
HPAT, 107 ,2P,, 122, 123
HPAT,  94 ,2P,, 107, 108
HPAT, 101 ,2P,, 115, 116
HPAT, 108 ,2P,, 123, 124
HPAT,  95 ,2P,, 108, 109
HPAT, 102 ,2P,, 116, 117
HPAT, 109 ,2P,, 124, 125
HPAT,  96 ,2P,, 109, 110
HPAT, 103 ,2P,, 117, 118
HPAT, 110 ,2P,, 125, 126
HPAT,  97 ,2P,, 110, 111
HPAT, 104 ,2P,, 118, 119
HPAT, 111 ,2P,, 126, 127
HPAT,  98 ,2P,, 111, 112
HPAT, 105 ,2P,, 119, 120
HPAT, 112 ,2P,, 127, 128
 SET,LINES,0
 NAME,GE 1
 SET,ACTIVE,NONE
 NAME,PI 1,PL
MESH,H   1T  35,HEX,N,1/   4/1/   4/1
MESH,H  36T  49,HEX,N,   2/   4/   2/   4/1
MESH,H  50T  56,HEX,N,3/   2/3/   2/1
 NAME,PIN
 SET,ACTIVE,NONE
 NAME,GE 1,PLOT
MESH,H  57T  91,HEX,N,1/   4/1/   4/1
MESH,H  92T 105,HEX,N,   4/   2/   4/   2/1
MESH,H 106T 112,HEX,N,   2/3/   2/3/1
```

```
NAME,GEAR
GR,1T#,DEL
NAME,PINION,RO,4(0)/1/0/-90,PIN
NAME,PINION,PL
```

TYPICAL FE MESH




Y
Z
X

```
C                               POINTS.F
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C       POINTS.F GENERATES N BY M POINTS FOR THE PINION & GEAR SURFACES
C                         POINTS.OUT => N BY M POINTS
C                               (MAIN PROGRAM)
C
        COMMON/CONST/PI,R,Q,MU,DEDEN,PSI,S,MCW,LM,EM,INC
        DIMENSION XYZ(4),U(51),THETA(51),PHIC(51),D(3,3),F(3),Y(3)
        DIMENSION RBAR(50,50),ZBAR(50,50),X1(50,50),X2(50,50),X3(50,50)
        DOUBLE PRECISION MU,A1,B1,C1,D1,E1,AA,BB,CC,RV,MCW,Q,PSI,PI
        DOUBLE PRECISION THETA,D,F,Y,GAMMA,DEDEN,R,S,TAU,EM,LM,PHIC
        DOUBLE PRECISION XYZ,U,INC,ZBAR,RBAR,X1,X2,X3,LV,ADDAN,CL,RL,FW
        INTEGER N1,N2,N3,M,UU,T,INT,BBB,CCC,SS
C
        WRITE(*,'(''PLEASE ENTER THE GRID PATTERN REQUIRED.''/
     *''FOR EXAMPLE: FOR A 7X8 PATTERN, ENTER 7 AND RETURN''/
     *''THEN ENTER 8 '')')
C
        READ(*,*)BBB
        READ(*,*)CCC
C
                PI     = 4.0 * ATAN(1.0)
                INC    = 0.1
                N1     = 11
                N2     = 3
                N3     = 4
        DO 5 INT = 1,4
C
       IF (INT .EQ. 1) THEN
C
C
C
C
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C                     DESCRIPTION OF INPUT DATA
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C
C                 1) RADIUS OF CUTTER, INCHES             (R)
C                 2) CRADLE ANGLE, DEGREES                (Q)
C                 3) BLADE ANGLE, DEGREES                 (PSI)
C                 4) CRADLE TO CUTTER DISTANCE, INCHES (S)
C                 5) RATIO OF ROLL                        (MCW)
C                 6) MACHINE OFFSET, INCHES               (EM)
C                 7) VECTOR SUM, INCHES                   (LM)
C                 8) DEDENDUM ANGLE, DEGREES              (DEDEN)
C                 9) PITCH ANGLE, DEGREES                 (MU)
C                10) ADDENDUM ANGLE, DEGREES              (ADDAN)
C                11) CLEARANCE, INCHES                    (CL)
C                12) MEAN CONE DISTANCE, INCHES           (RL)
C                13) FACE WIDTH, INCHES                   (FW)
C                14) INIITIAL GUESS FOR SURFACE COORDINATE U      ( U(1) )
C                15) INITIAL GUESS FOR SURFACE COORDINATE THETA  ( THETA(1) )
C                16) INITIAL GUESS FOR ANGLE OF CRADLE, DEGREES  ( PHIC(1)  )
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C             INSERT CONCAVE SIDE OF PINION DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C
               R      = 2.96562137806
               Q      = 63.9420304635 * PI/180.0
               PSI    = 161.954330248 * PI/180.0
               S      = 2.94780202969
               MCW    = 0.30838512709
```

```
             EM    = 0.154575896
             LM    = 0.0384999977874
             DEDEN = 1.56666666 * pi/180
             MU    = 18.4333333 * pi/180
             ADDAN = 3.8833334  * pi/180
             CL    = 0.03
             RL    = 3.191
             FW    = 1.0
             U(1)  = 9.59703
          THETA(1) = 126.83544 * PI/180.0
          PHIC(1)  = -0.85813 * PI/180.0
C
           ELSEIF (INT .EQ. 2) THEN
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C             INSERT CONVEX SIDE OF PINION DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
               R     = 3.071306157
               Q     = 53.9259945467 * PI/180.0
               PSI   = 24.337423854 * PI/180.0
               S     = 2.80104946
               MCW   = 0.3220428536
               EM    = -0.17426159493
               LM    = -0.0518138227
               DEDEN = 1.56666666 * pi/180
               MU    = 18.4333333 * pi/180
               ADDAN = 3.8833334  * pi/180
               CL    = 0.03
               RL    = 3.191
               FW    = 1.0
               U(1)  = 7.42534
            THETA(1) = 124.43689* PI/180.0
            PHIC(1)  = -11.38663*PI/180.0
C
           ELSEIF (INT .EQ. 3) THEN
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C             INSERT CONCAVE SIDE OF GEAR DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
             R     = 3.0325
             Q     = 59.2342023 * PI/180.0
             PSI   = 158.0 * PI/180.0
             S     = 2.85995004691
             MCW   = 0.9508646
             EM    = 0.0
             LM    = 0.0
             DEDEN = 3.8833333333 * pi/180
             MU    = 71.5666666    * pi/180
             ADDAN = 1.5666666    * pi/180
             CL    = 0.0366
             RL    = 3.191
             FW    = 1.0
             U(1)  = 8.12602
          THETA(1) = 233.98994 * PI/180.0
          PHIC(1)  = -0.35063 * PI/180.0
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C
C
C
C
           ELSE
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C             INSERT CONVEX SIDE OF GEAR DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
               R      = 2.9675
               Q      = 59.2342023 * PI/180.0
               PSI    = 22.0 * PI/180.0
               S      = 2.85995004691
               MCW    = 0.9508646
               EM     = 0.0
               LM     = 0.0
               DEDEN  = 3.8833333333 * pi/180
               MU     = 71.5666666    * pi/180
               ADDAN  = 1.5666666    * pi/180
               CL     = 0.0366
               RL     = 3.191
               FW     = 1.0
               U(1)   = 7.89156
            THETA(1)  = 234.95451 * PI/180.0
            PHIC(1)   = -12.3384 * PI/180.0
        ENDIF
C
        OPEN(UNIT=900,FILE='POINTS.OUT',STATUS='UNKNOWN')
C
        DO 10 UU = 1,BBB
        DO 20 T  = 1,CCC
C
        CALL STEPZR(ZBAR,RBAR,UU,T,int,BBB,CCC,deden,mu,addan,cl,rl,fw)
C
        SS = BBB - 1
        DO 100 M = 1,SS
C
        CALL DIFF(U,THETA,PHIC,M,D,N1,N2,UU,T,ZBAR,RBAR,INT)
C
        IF ((INT .EQ. 1).OR.(INT .EQ. 2)) THEN
             TAU   = (THETA(M)) - Q + (PHIC(M))
             GAMMA = MU - DEDEN
C
        ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN
             TAU   = (THETA(M)) + Q - (PHIC(M))
             GAMMA = MU - DEDEN
        ENDIF
C
C   (THE FIRST EQUATION. (EQUATION OF MESHING FOR LEFT HAND PINION))
C
           A1 = ((U(M))-R*(COS(PSI)*COS(PSI)/SIN(PSI)))
     1           *COS(GAMMA)*SIN(TAU)
           B1 = S*(MCW-SIN(GAMMA))*COS(PSI)*SIN(THETA(M))
           C1 = S*COS(GAMMA)*SIN(PSI)*SIN(Q-(PHIC(M)))
           D1 = EM*(COS(GAMMA)*SIN(PSI)+SIN(GAMMA)*COS(PSI)*COS(TAU))
           E1 = LM*SIN(GAMMA)*COS(PSI)*SIN(TAU)
        IF ((INT .EQ. 1).OR.(INT .EQ. 2)) THEN
           LV = A1 + B1 - C1 + D1 - E1
           AA = LV
        ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN

           RV = A1 + B1 + C1 - D1 - E1
           AA = RV
        ENDIF
C
C            (THE SECOND EQUATION.  ZW  -  Z  =  0)
C
        CALL TRANSF(U,THETA,PHIC,M,XYZ,N3,N1,INT)
C
           BB = XYZ(3) - ZBAR(UU,T)
C
C         (THE THIRD EQUATION.   R  -  SQRT( X*X  +  Y*Y )  =  0)
C
           CC = RBAR(UU,T) - SQRT(XYZ(1)*XYZ(1) + XYZ(2)*XYZ(2))
           F(1) = -AA
           F(2) = -BB
           F(3) = -CC
C
        CALL GAUSS(D,F,Y,N2)
C
           U(M+1)      = U(M) + Y(1)
           THETA(M+1) = THETA(M) + Y(2)
           PHIC(M+1)  = PHIC(M) + Y(3)
 100    CONTINUE
C
            X1(UU,T) = XYZ(1)
            X2(UU,T) = XYZ(2)
            X3(UU,T) = XYZ(3)
C
        WRITE(900,*)X1(UU,T),X2(UU,T),X3(UU,T)
C
 20     CONTINUE
 10     CONTINUE
C
 5      CONTINUE
        CLOSE(900,STATUS='KEEP')
C
        STOP
        END
C
        SUBROUTINE DIFF(X1,X2,X3,M,D,N1,N2,UU,T,ZBAR,RBAR,INT)
C
C        (THE VARIABLES X1,X2,X3; LOCAL TO THIS PROCEDURE;REPRESENT
C              U, THETA AND PHIC.    )
C
        COMMON/CONST/PI,R,Q,MU,DEDEN,PSI,S,MCW,LM,EM,INC
        DIMENSION X1(N1),X2(N1),X3(N1),A(5,3),B(5,3),C(5,3),RVAL(5,3)
        DIMENSION XX1(51),XX2(51),XX3(51),D(N2,N2),XYZ(4),ZBAR(50,50)
        DIMENSION RBAR(50,50),LVAL(5,3)
        INTEGER I,J,L,UU,T
        DOUBLE PRECISION RVAL,A,B,C,K,H1,H2,H3,TAU,GAMMA,A1,B1,C1,D1
        DOUBLE PRECISION MU,DEDEN,PI,LM,MCW,X1,X2,X3,XX1,XX2,XX3,D
        DOUBLE PRECISION E1,INC,R,PSI,XYZ,EM,Q,S,ZBAR,RBAR,LVAL
C
C              (H1,H2,H3 ARE THE INCREMENTS ADDED TO THE)
C              (VAR X1,X2,X3 DURING THE NUMERICAL DIFF. )
C       (A1,B1,C1,D1,E1 INTERMEDIATE VALUES FOR EQU. OF MESHING)
C
        DO 201 I = 1,5
                L = I - 3
                K = L/2.0
C
C             (K IS THE MULTIPLIER ON THE INCREMENT "INC".)
C
             DO 205 J = 1,3
C
         IF (J .EQ. 1) THEN
               H1 =INC
               H2 = 0.
               H3 = 0.
         ELSE IF (J .EQ. 2) THEN
               H1 = 0.
               H2 = INC
               H3 = 0.
         ELSE
               H1 = 0.
               H2 = 0.
               H3 = INC
         ENDIF
C
```

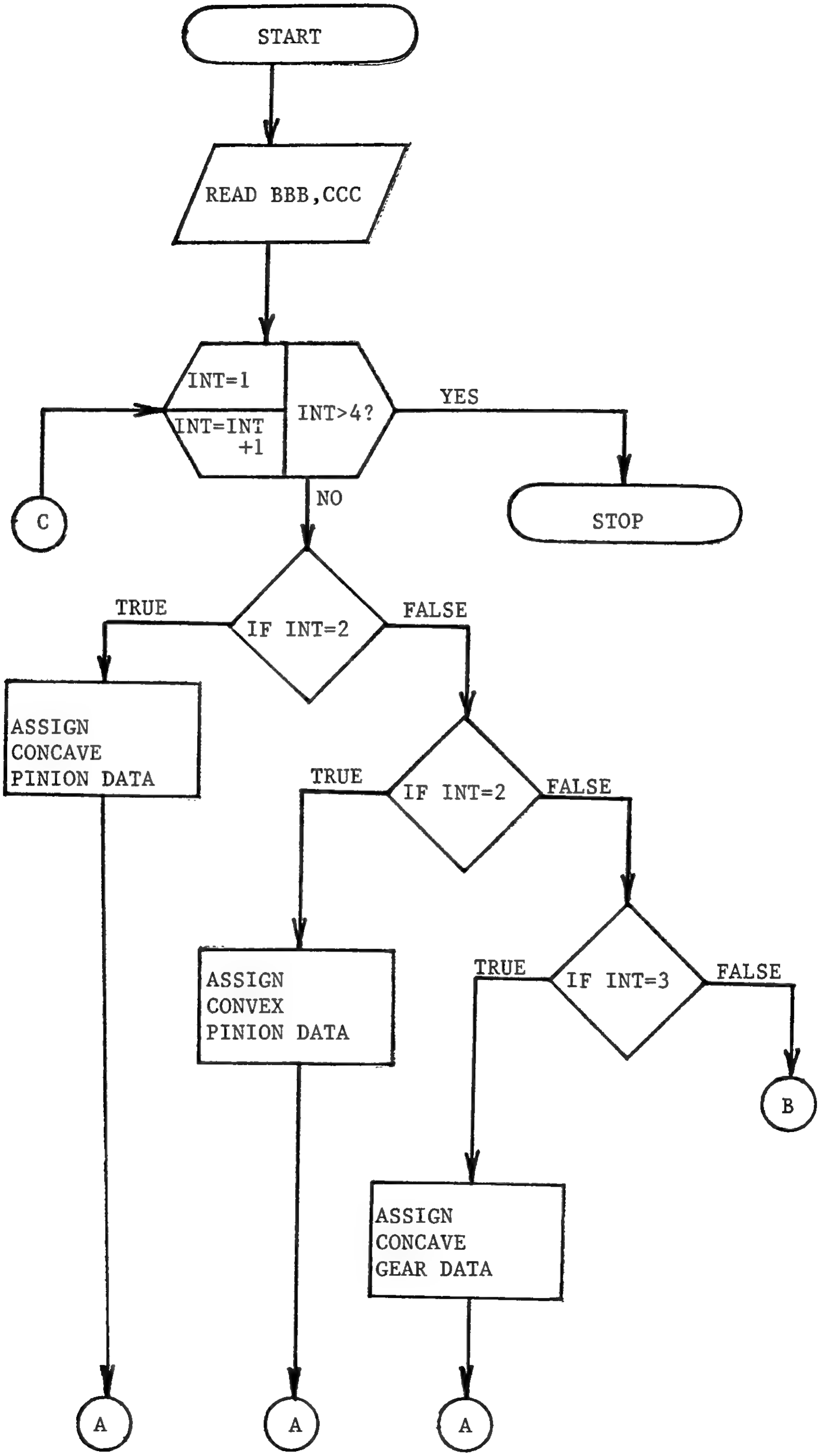
```
C       (* INSERT THE THREE EQUATIONS TO BE DIFFERENTIATED HERE *)
C        ( ADD K*H1, K*H2, K*H3 TO EACH VARIABLE X1,X2,X3 )
C
         IF ((INT .EQ. 1).OR.(INT .EQ. 2)) THEN
            TAU   = (X2(M)+K*H2) - Q + (X3(M)+K*H3)
            GAMMA = MU - DEDEN
         ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN
            TAU   = (X2(M)+K*H2) + Q - (X3(M)+K*H3)
            GAMMA = MU - DEDEN
         ENDIF
C
C       (THE FIRST EQUATION. (EQUATION OF MESHING FOR LEFT HAND PINION))
C
           A1 = ((X1(M)+K*H1)-R*(COS(PSI)*COS(PSI)/SIN(PSI)))
     *          *COS(GAMMA)*SIN(TAU)
           B1 = S*(MCW-SIN(GAMMA))*COS(PSI)*SIN(X2(M)+K*H2)
           C1 = S*COS(GAMMA)*SIN(PSI)*SIN(Q-(X3(M)+K*H3))
           D1 = EM*(COS(GAMMA)*SIN(PSI) + SIN(GAMMA)*COS(PSI)*COS(TAU))
           E1 = LM*SIN(GAMMA)*COS(PSI)*SIN(TAU)
         IF ((INT .EQ. 1).OR.(INT .EQ. 2)) THEN
           LVAL(I,J) = A1 + B1 - C1 + D1 - E1
           A(I,J)    = LVAL(I,J)
         ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN
           RVAL(I,J) = A1 + B1 + C1 - D1 - E1
           A(I,J)    = RVAL(I,J)
         ENDIF
C
          (THE SECOND EQUATION.  ZW - Z = 0)
C
            XX1(M) = X1(M) + K*H1
            XX2(M) = X2(M) + K*H2
            XX3(M) = X3(M) + K*H3
C
        CALL TRANSF(XX1, XX2, XX3,M,XYZ,N3,N1,INT)
C
            B(I,J) = XYZ(3) - ZBAR(UU,T)
C
C       (THE THIRD EQUATION.  R - SQRT(X*X + Y*Y) = 0)
C
            C(I,J) = RBAR(UU,T) - SQRT(XYZ(1)*XYZ(1) + XYZ(2)*XYZ(2))
 205       CONTINUE
C
 201    CONTINUE
C
        DO 210 I = 1,3
            D(1,I) = -(A(5,I) - 8*A(4,I) - A(1,I) + 8*A(2,I))/(6*INC)
            D(2,I) = -(B(5,I) - 8*B(4,I) - B(1,I) + 8*B(2,I))/(6*INC)
            D(3,I) = -(C(5,I) - 8*C(4,I) - C(1,I) + 8*C(2,I))/(6*INC)
 210    CONTINUE
C
        RETURN
        END
C
C
        SUBROUTINE TRANSF( X1,X2,X3,M,XYZ,N3,N1,INT)
C
        COMMON/CONST/PI,R,Q,MU,DEDEN,PSI,S,MCW,LM,EM,INC
        DIMENSION X1(N1),X2(N1),X3(N1),A(4),REST1(4,4)
        DIMENSION REST2(4,4),REST3(4,4),REST4(4,4),MSC(4,4),MMS(4,4)
        DIMENSION MPM(4,4),MAP(4,4),MWA(4,4),XYZ(N3)
        DOUBLE PRECISION INC,PI,MCW,Q,LM,MU,PSI,MSC,MMS,MPM,MAP,MWA
        DOUBLE PRECISION REST1,REST2,REST3,REST4,DEDEN,S,R,EM,PHIW
        DOUBLE PRECISION X1,X2,X3,A,XYZ
        INTEGER I,J,K
C
        DO 300 I = 1,4
```

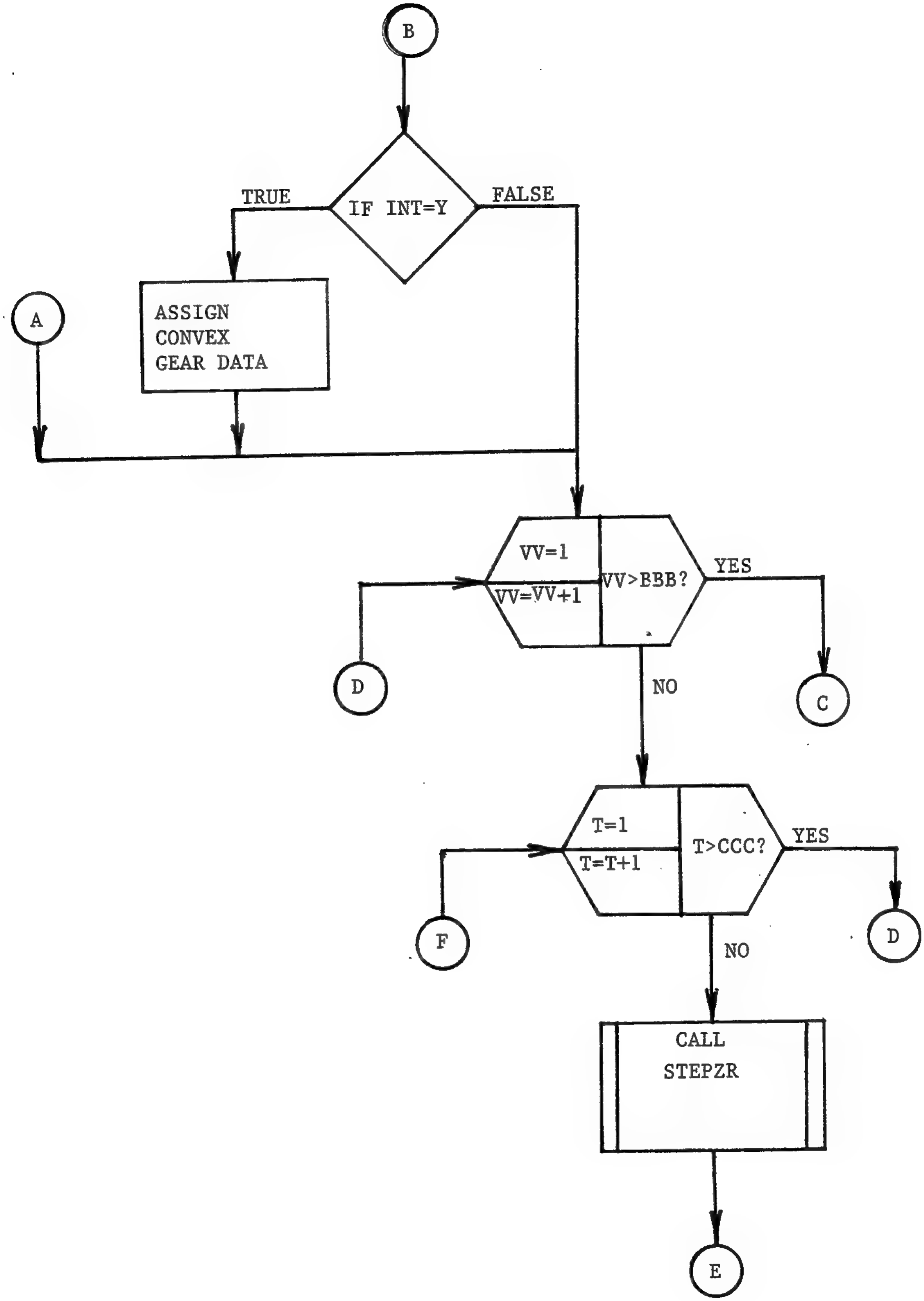
```
C
        DO 310 J = 1,4
               REST1(I,J)=0
               REST2(I,J)=0
               REST3(I,J)=0
               REST4(I,J)=0
               MSC(I,J)=0
               MMS(I,J)=0
               MPM(I,J)=0
               MAP(I,J)=0
               MWA(I,J)=0
 310    CONTINUE
C
 300    CONTINUE
C
                  PHIW  =  X3(M)/MCW
C
C       ( THE COORDINATE TRANSFORMATIONS )
C
         IF ((INT .EQ. 1).OR.(INT .EQ. 2)) THEN
          MSC(1,1) = 1.0
          MSC(2,2) = COS(Q)
          MSC(2,3) = -SIN(Q)
          MSC(2,4) = -S*SIN(Q)
          MSC(3,2) = +SIN(Q)
          MSC(3,3) = COS(Q)
          MSC(3,4) = S*COS(Q)
          MSC(4,4) = 1.0
C
          MPM(1,1) = COS(DEDEN)
          MPM(1,3) = -SIN(DEDEN)
          MPM(1,4) = -LM*SIN(DEDEN)
          MPM(2,2) = 1.0
          MPM(2,4) = +EM
          MPM(3,1) = SIN(DEDEN)
          MPM(3,3) = COS(DEDEN)
          MPM(3,4) = LM*COS(DEDEN)
          MPM(4,4) = 1.0
C
          MMS(1,1) = 1.0
          MMS(2,2) = COS(X3(M))
          MMS(2,3) = +SIN(X3(M))
          MMS(3,2) = -SIN(X3(M))
          MMS(3,3) = COS(X3(M))
          MMS(4,4) = 1.0
C
          MAP(1,1) = COS(MU)
          MAP(1,3) = SIN(MU)
          MAP(2,2) = 1.0
          MAP(3,1) = -SIN(MU)
          MAP(3,3) = COS(MU)
          MAP(4,4) = 1.0
C
          MWA(1,1) = COS(PHIW)
          MWA(1,2) = +SIN(PHIW)
          MWA(2,1) = -SIN(PHIW)
          MWA(2,2) = COS(PHIW)
          MWA(3,3) = 1.0
          MWA(4,4) = 1.0
C
         ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN
C
          MSC(1,1) = 1.0
          MSC(2,2) = COS(Q)
          MSC(2,3) = +SIN(Q)
          MSC(2,4) = +S*SIN(Q)
          MSC(3,2) = -SIN(Q)
          MSC(3,3) = COS(Q)
          MSC(3,4) = S*COS(Q)
          MSC(4,4) = 1.0
C
          MPM(1,1) = COS(DEDEN)
          MPM(1,3) = -SIN(DEDEN)
          MPM(1,4) = -LM*SIN(DEDEN)
          MPM(2,2) = 1.0
          MPM(2,4) = -EM
          MPM(3,1) = SIN(DEDEN)
          MPM(3,3) = COS(DEDEN)
          MPM(3,4) = LM*COS(DEDEN)
          MPM(4,4) = 1.0
C
          MMS(1,1) = 1.0
          MMS(2,2) = COS(X3(M))
          MMS(2,3) = -SIN(X3(M))
          MMS(3,2) = +SIN(X3(M))
          MMS(3,3) = COS(X3(M))
          MMS(4,4) = 1.0
C
          MAP(1,1) = COS(MU)
          MAP(1,3) = SIN(MU)
          MAP(2,2) = 1.0
          MAP(3,1) = -SIN(MU)
          MAP(3,3) = COS(MU)
          MAP(4,4) = 1.0
C
          MWA(1,1) = COS(PHIW)
          MWA(1,2) = -SIN(PHIW)
          MWA(2,1) = +SIN(PHIW)
          MWA(2,2) = COS(PHIW)
          MWA(3,3) = 1.0
          MWA(4,4) = 1.0
C
         ENDIF
C
C       ( THE MATRIX MULTIPLICATIONS )
C
          DO 320 I = 1,4
            DO 325 J = 1,4
               DO 330 K = 1,4
                 REST1(I,J)  = REST1(I,J)  + MMS(I,K)*MSC(K,J)
 330           CONTINUE
 325        CONTINUE
 320      CONTINUE
C
          DO 335 I = 1,4
            DO 340 J = 1,4
               DO 345 K = 1,4
                 REST2(I,J)  = REST2(I,J)  + MPM(I,K)*REST1(K,J)
 345           CONTINUE
 340        CONTINUE
 335      CONTINUE
C
          DO 350 I = 1,4
            DO 355 J = 1,4
               DO 360 K = 1,4
                 REST3(I,J)  = REST3(I,J)  + MAP(I,K)*REST2(K,J)
 360           CONTINUE
 355        CONTINUE
 350      CONTINUE
C
          DO 365 I = 1,4
            DO 370 J = 1,4
```

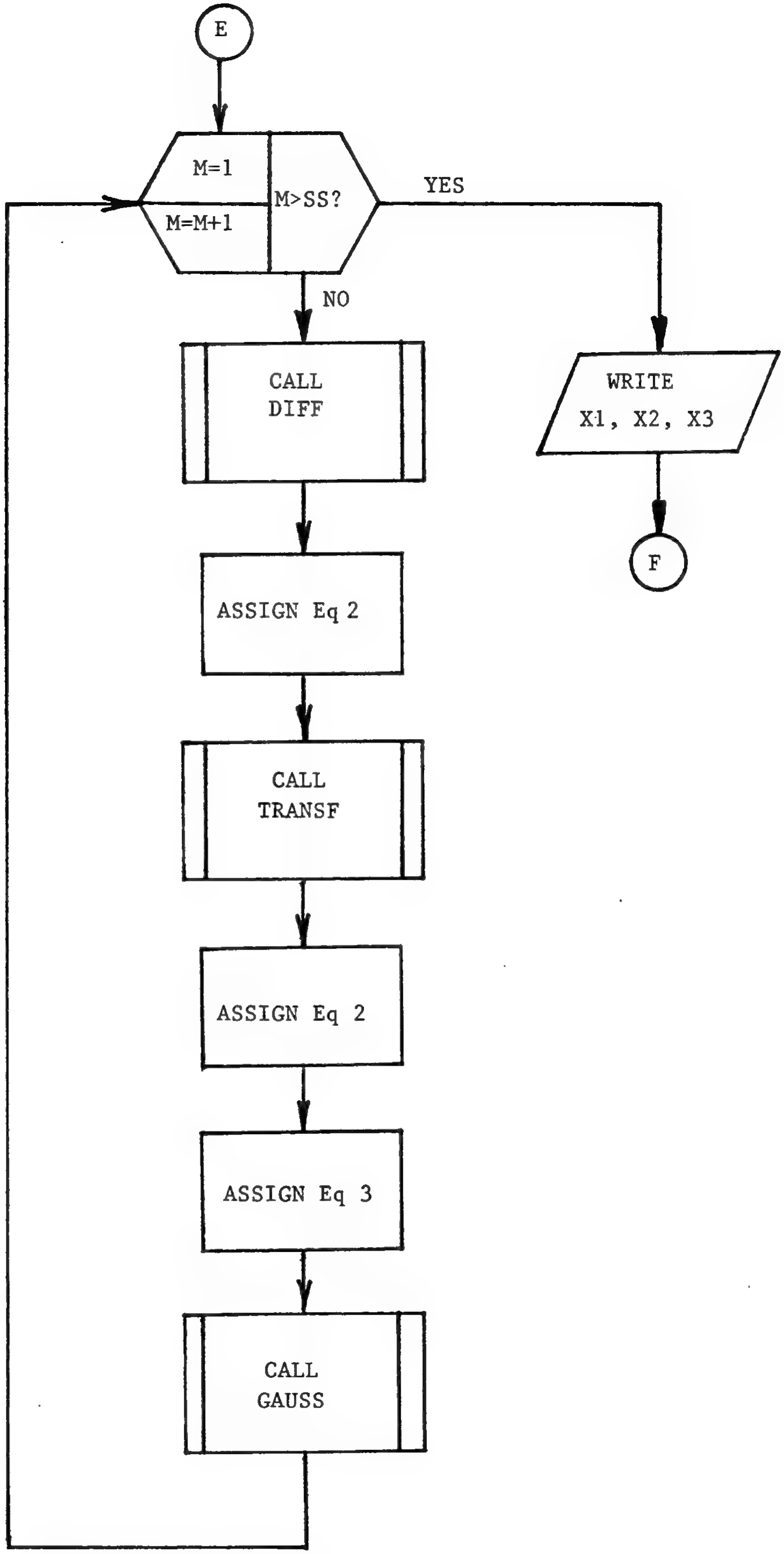
```
               DO 375 K = 1,4
                 REST4(I,J)  = REST4(I,J)  + MWA(I,K)*REST3(K,J)
 375           CONTINUE
 370        CONTINUE
 365      CONTINUE
C
          DO 380 I = 1,4
            XYZ(I) = 0
 380      CONTINUE
C
             A(1) =  R*COS(PSI)/SIN(PSI)-X1(M)*COS(PSI)
             A(2) =  X1(M)*SIN(PSI)*SIN(X2(M))
             A(3) =  X1(M)*SIN(PSI)*COS(X2(M))
             A(4) =  1.0
C
         DO 385 K = 1,4
           DO 390 I = 1,4
             XYZ(K) = XYZ(K) + REST4(K,I)*A(I)
 390       CONTINUE
 385     CONTINUE
C
        RETURN
        END
C
C
        SUBROUTINE  GAUSS(D,F,Y,N2)
C
        DIMENSION D(N2,N2),F(N2),Y(N2)
        DOUBLE PRECISION  PIVOT,MULT,TOP,D,F,Y
        INTEGER I,J,K,N
C
        N = 3
C
        DO 400 J = 1,N-1
        PIVOT = D(J,J)
C
        DO 410 I = J+1,N
            MULT = D(I,J)/PIVOT
C
           DO 420 K = J+1,N
            D(I,K) = D(I,K) - MULT * D(J,K)
              F(I) = F(I) - MULT * F(J)
 420      CONTINUE
C
 410    CONTINUE
C
             Y(N) = F(N)/D(N,N)
        DO 430 I = N-1,1,-1
             TOP = F(I)
           DO 440 K = I+1,N
                TOP  = TOP - D(I,K) * Y(K)
                Y(I) = TOP/D(I,I)
 440       CONTINUE
C
 430    CONTINUE
C
 400    CONTINUE
C
        RETURN
        END
C
C
        Subroutine stepzr(zbar,rbar,uu,t,int,bbb,ccc,deden,mu,addan
     * ,cl,rl,fw)
C
        DIMENSION ZBAR(50,50),RBAR(50,50)
```

```
      DOUBLE PRECISION ZPITCH,ZROOT1,ZM1,ZMX,ZINC1,ZM,RM,G,G9,Z,R
      DOUBLE PRECISION ZBAR,RBAR,DEDEN,ADDAN,MU,CL,RL,FW
      INTEGER DZ1,DR1,UU,T,BBB,CCC
            PI      = 4.0*ATAN(1.0)
C
      ZPITCH = RL - FW/2.0
      ZROOT1 = ZPITCH * COS(DEDEN)
      ZM1    = ZROOT1 * COS(MU - DEDEN)
      ZMX    = (ZROOT1 + FW) * COS(MU - DEDEN)
      ZINC1  = (ZMX - ZM1)/(BBB - 1)
      Z      = 0.
      R      = 0.
C
      DZ1    = (UU - 1)
      ZM     = ZM1 + DZ1*ZINC1 - CL * SIN(MU -DEDEN)
      RM     = ZM * TAN(MU - DEDEN) + CL/COS(MU - DEDEN)
      G      = ZM * TAN(ADDAN + DEDEN)/COS(MU - DEDEN) - CL
      G9     = G/(ccc - 1)
C
C
              dr1 = (T-1)
              Zbar(uu,t)  = zm - dr1*g9*sin(mu-deden)
              rbar(uu,t)  = rm + dr1*g9*cos(mu-deden)
      RETURN
      END
```

FLOWCHART: POINTS.F




B
TRUE
IF INT=Y
FALSE
A
ASSIGN
CONVEX
GEAR DATA
VV=1
VV=VV+1
VV>BBB?
YES
D
NO
C
T=1
T=T+1
T>CCC?
YES
F
NO
D
CALL
STEPZR
E


E
M=1
M=M+1
M>SS?
YES
NO
CALL DIFF
WRITE X1, X2, X3
F
ASSIGN Eq 2
CALL TRANSF
ASSIGN Eq 2
ASSIGN Eq 3
CALL GAUSS

FLOWCHART: POINTS.F (DISCUSSION)

There are 4 loops in the main program.

Loop 1: loops one through four to read data for the four surfaces.

Loop 2: loops one through BBB, where BBB is the number of points across the face of the tooth surface.

Loop 3: loops one through CCC, where CCC is the number of points along the length of the tooth height.

Loop 4: The final loop iterates to solve the three equations used to identify a point on the surface.

Subroutines:

1. STEPZR. This subroutine steps Z and R as the solution "marches" across the tooth face.

2. DIFF. This subroutine performs the numerical differentiation of the three equations (to form the Jacobian).

3. TRANSF. Subroutine that performs the matrix multiplication for the five coordinate transformations.

4. GAUSS. Uses gauss elimination to solve the Jacobian matrix.

```
C                               PAT.F
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C       THIS PROGRAM GIVES THE N * M GRID PATRAN INPUT FILE FOR     C
C     THE GENERATION OF SPIRAL BEVEL GEAR TOOTH SURFACE WITH FILLET   C
C-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=C
C             READS POINTS.OUT AND GIVES T1.OUT
C                  T1.OUT => INPUT FILE FOR PATRAN
C
C                          (MAIN PROGRAM)
C
      COMMON/UNITS/NF1,NF2
      DIMENSION X1(45,45),X2(45,45),X3(45,45)
      DIMENSION X1R(45,45),X2R(45,45),XX(8000),ZZ(8000),YY(8000)
      DOUBLE PRECISION ri,cl,r1,r2,mu,deden,rogear
      DOUBLE PRECISION X1,X2,X3,X1R,X2R,XX,YY,ZZ,rotcon,rotint
      INTEGER UU,T,INT,NF1,NF2,NF3,JI,BB,CC
      INTEGER NUMBER,FILL,NTPIN,NTGE
C
               PI    = 4.0*ATAN(1.0)
C
      NF1 = 900
      NF2 = 1000
      NF3 = 1001
C
      OPEN(UNIT=NF1,FILE='t1.out',STATUS='UNKNOWN')
      OPEN(UNIT=NF2,FILE='t2.out',STATUS='UNKNOWN')
      OPEN(UNIT=NF3,FILE='POINTS.OUT',STATUS='OLD')
C
      WRITE(*,'(''PLEASE ENTER THE REQUIRED GRID PATTERN.''/
     *''FOR EXAMPLE : FOR A 8X7 PATTERN ENTER 8 AND THEN 7 >'')')
      READ(*,*)BB
      READ(*,*)cc
      write(*,*)'ENTER NUMBER OF ELEMENTS THROUGH THE THICKNESS'
        READ(*,*) number
        WRITE(*,*)'ENTER THE NUMBER OF ELEMENTS IN THE FILLET'
        READ(*,*)FILL
      DO 5 INT = 1,4
        IF ((INT .EQ. 1).OR.(INT .EQ. 2)) THEN
C
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C              DESCRIPTION OF INPUT DATA FOR PINION
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C        1. DEDENDUM, DEGREES                      (DEDEN)
C        2. PITCH ANGLE, DEGREES                   (MU)
C        3. ROTATION OF CONVEX SURFACE TO
C                CREATE TOP LAND                   (ROTCON)
C        4. ROTATION OF PINION TO ELIMINATE
C                INTERFERENCE                      (ROTINT)
C        5. PINION ID, INCHES                      (RI)
C        6. CLEARANCE, INCHES                      (CL)
C        7. NUMBER OF PINION TEETH                 (NTPIN)
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C           INPUT THE PINION DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
              DEDEN = 1.56666666 * PI/180.0
               MU   = 18.4333333 * PI/180.0
               ROTCON = 2.275
               ROTINT = -3.56
               RI   = 0.609375
               CL   = 0.03
               NTPIN = 12
C-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C
C
        ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN
C
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C       DESCRIPTION OF INPUT DATA FOR THE GEAR
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C        1. DEDENDUM, DEGREES                           (DEDEN)
C        2. PITCH ANGEL, DEGREES                        (MU)
C        3. ROTATION OF CONVEX SURFACE TO
C              CREATE TOP LAND                          (ROTCON)
C        4. ROTATION OF GEAR TO PUT IN MESH             (ROGEAR)
C        5. CLEARANCE, INCHES                           (CL)
C        6. ID OF GEAR BASE                             (R1)
C        7. OD OF GEAR BASE                             (R2)
C        8. NUMBER OF GEAR TEETH                        (NTGE)
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C           INPUT THE GEAR DATA BELOW
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
             DEDEN = 3.883333333 * PI/180.0
              MU    = 71.5666666 * PI/180.0
              ROTCON = -8.49
              ROGEAR = 190.0
              CL = 0.0366
              R1 = 2.375
              R2 = 3.250
              NTGE = 36
C=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
C
C
C
C
C
C
        ENDIF
C
        IF (int.EQ.1) THEN
            WRITE(NF1,*)'SET,LABEL,OFF'
            WRITE(NF1,*)'VI'
            WRITE(NF1,*)'1'
            WRITE(NF1,*)'120,0,120'
        ENDIF
C
        JI = 1
C
      DO 10 UU = 1,BB
      DO 20 T  = 1,cc
C
      READ(NF3,*)X1(UU,T),X2(UU,T),X3(UU,T)
C
      IF ((INT .EQ. 2).OR.(INT .EQ. 4)) THEN
       CALL ROTATE(X1(UU,T),X2(UU,T),X1R(UU,T),X2R(UU,T),INT,
     * rotcon)
      ENDIF
C
      CALL ALLIGN(X1(UU,T),X2(UU,T),X1R(UU,T),X2R(UU,T),INT,
     * rotint,rogear)
C
      CALL GRID(UU,T,INT,X1,X2,X3,X1R,X2R,XX,YY,ZZ,bb,cc)
C
        JI = JI + 1
```

```
C
 20     CONTINUE
 10     CONTINUE
 5      CONTINUE
C
        DO 6 int = 1,4
C
        IF ((INT .EQ. 2).OR.(INT .EQ. 4)) THEN
           CALL LINE(INT,bb,cc)
           SS = INT
           CALL PATCH(SS,bb,cc)
           CALL HPAT(INT,bb,cc)
           CALL FILLET(XX,YY,ZZ,INT,bb,cc,ri,cl,r1,r2,mu,deden,
     * ntpin,ntge)
        ENDIF
C
 6      CONTINUE
C
         CALL MODEL(bb,cc,number,fill)
        CLOSE(NF3,STATUS='KEEP')
        CLOSE(NF2,STATUS='KEEP')
        CLOSE(NF1,STATUS='KEEP')
C
        STOP
        END
C
C
      SUBROUTINE ROTATE(X1,X2,X1R,X2R,INT,rotcon)
C
        COMMON/UNITS/NF1,NF2
        DOUBLE PRECISION PI,Q,A,X1,X2,X1R,X2R,rotcon
        DIMENSION A(2,2)
C
C     ROTATION OF PINION AND GEAR CONVEX SIDES BY Q DEG. TO CREATE TOP LAND
C
             PI   = 4.0 * ATAN(1.0)
        IF ((INT .EQ. 1).OR.(INT .EQ. 2)) THEN
             Q      = rotcon*PI/180.0
        ELSEIF ((INT .EQ. 3).OR.(INT .EQ. 4)) THEN
             Q      = rotcon*PI/180.0
        ENDIF
           A(1,1) = COS(Q)
           A(2,1) = -SIN(Q)
           A(1,2) = SIN(Q)
           A(2,2) = COS(Q)
C
        X1R = X1*A(1,1)+X2*A(2,1)
        X2R = X1*A(1,2)+X2*A(2,2)
C
        RETURN
        END
C
C
       SUBROUTINE ALLIGN(X1,X2,X1R,X2R,INT,rotint,rogear)
C
C     TO ALLIGN THE PINION ABOVE THE GEAR AND CALCULATE GAPS
C
      COMMON/UNITS/NF1,NF2
      DOUBLE PRECISION X1,X2,X1R,X2R,X11,X22,X11R,X22R
      DOUBLE PRECISION Q,PI,X18,Y18,X18R,Y18R,rotint,rogear
C
      PI = 4.0 * ATAN(1.0)
C
C     ROTATION OF THE PINION BY -3.56 DEG. ABOUT Z-AXIS
C
      Q  =rotint*PI/180.0
      IF(INT.EQ.1)THEN
            X11 = X1*COS(Q)-X2*SIN(Q)
            X22 = X1*SIN(Q)+X2*COS(Q)
            X1  = X11
            X2  = X22
      ELSEIF(INT.EQ.2)THEN
            X11R = X1R*COS(Q)-X2R*SIN(Q)
            X22R = X1R*SIN(Q)+X2R*COS(Q)
            X1R  = X11R
            X2R  = X22R
      ENDIF
C
      Q  =-30.0*PI/180.0
      IF(INT.EQ.1)THEN
            X30 = X1*COS(Q)-X2*SIN(Q)
            Y30 = X1*SIN(Q)+X2*COS(Q)
      ELSEIF(INT.EQ.2)THEN
            X30R = X1R*COS(Q)-X2R*SIN(Q)
            Y30R = X1R*SIN(Q)+X2R*COS(Q)
      ENDIF
C
C     ROTATION OF THE GEAR BY 190 DEG. ABOUT Z-AXIS
C
      Q  = rogear*PI/180.0
      IF(INT.EQ.3)THEN
            X18 = X1*COS(Q)-X2*SIN(Q)
            Y18 = X1*SIN(Q)+X2*COS(Q)
            X1  = X18
            X2  = Y18
      ELSEIF(INT.EQ.4)THEN
            X18R = X1R*COS(Q)-X2R*SIN(Q)
            Y18R = X1R*SIN(Q)+X2R*COS(Q)
            X1R  = X18R
            X2R  = Y18R
      ENDIF
C
C
      RETURN
      END
C
C
         SUBROUTINE GRID(UU,T,INT,X1,X2,X3,X1R,X2R,XX,YY,ZZ,BB,cc)
C
         COMMON/UNITS/NF1,NF2
         INTEGER NN,UU,T,BB,cc
         DOUBLE PRECISION X1(45,45),X2(45,45),X3(45,45)
         DOUBLE PRECISION X1R(45,45),X2R(45,45)
         DOUBLE PRECISION XX(8000),YY(8000),ZZ(8000)
         IF (INT .EQ. 1) THEN
            NN = (UU - 1) * cc + T
         ELSEIF (INT .EQ. 2) THEN
            NN = (UU - 1) * cc + T + (BB*cc)
         ELSEIF (INT .EQ. 3) THEN
            NN = (UU - 1) * cc + T + 2*(BB*cc)
         ELSEIF (INT .EQ. 4) THEN
            NN = (UU - 1) * cc + T + 3*(BB*cc)
         ENDIF
C
         IF ((INT .EQ. 1).OR.(INT .EQ. 3)) THEN
               XX(NN) = X1(UU,T)
               YY(NN) = X2(UU,T)
               ZZ(NN) = X3(UU,T)
         ELSEIF ((INT .EQ. 2).OR.(INT .EQ. 4)) THEN
               XX(NN) = X1R(UU,T)
               YY(NN) = X2R(UU,T)
               ZZ(NN) = X3(UU,T)
```

```
        ENDIF
C
        WRITE(NF1,111)’GRID’,’,’,NN,’,,’,XX(NN),’/’,YY(NN),’/’,ZZ(NN)
 111    FORMAT(A4,A1,I4,A2,F10.6,A1,F10.6,A1,F10.6)
C
        RETURN
        END
C
        SUBROUTINE LINE(INT,bb,cc)
C
        COMMON/UNITS/NF1,NF2
        INTEGER L,M,N,K,bb,cc
C
        DO 600 L = 1,BB
               K = L
        IF (INT .EQ. 2) THEN
               M = (L-1) * cc + 1
               n = m+bb*cc
        ELSEIF (INT .EQ. 4) THEN
               K = L + BB*16
               m = (L - 1) *cc + (2*bb*cc + 1)
               n = M + bb*cc
        ENDIF
               N = M + BB*cc
        WRITE(NF1,610)’LINE’,’,’,K,’,’,’ST’,’,,’,M,’,’,N
        WRITE(nf1,611)’LINE,’,k+bb+L-1,’/’,k+bb+L,’,’,’BR,.5,’,k
  610   FORMAT(a4,a1,i4,a1,2a2,i4,a1,i4)
  611   FORMAT(a5,i3,a1,i3,a1,a6,i4)
  600   CONTINUE
C
        RETURN
        END
C
        SUBROUTINE PATCH(SS,bb,cc)
C
        COMMON/UNITS/NF1,NF2
        INTEGER I,II,J,JJ,K,L,M,bb,cc
C
        JJ = (cc-1)*BB
        DO 700 I = 1,JJ
C
        IF (SS .EQ. 2) THEN
               J  = INT((I-1)/(cc-1)) + I
               II = I
        ELSEIF (SS .EQ. 4) THEN
               j =  int((I-1)/(cc-1)) + I + 2*(bb*cc)
               II = jj + I + 3*bb
        ENDIF
               K  = J + 1
               L  = K + BB*cc
               M  = L - 1
        WRITE(NF1,710)’PATCH’,’,’,II,’,’,’QUAD’,’,,’,J,’/’,K,’/’,L,’/’,M
 710    FORMAT(A5,A1,I4,A1,A5,A2,I4,A1,I4,A1,I4,A1,I4)
 700    CONTINUE
C
        RETURN
        END
        SUBROUTINE HPAT (int,bb,cc)
C
        COMMON/units/nf1,nf2
        INTEGER i,j,bb,cc
C
        jj = (bb-1)*(cc-1)
        Do 800 I = 1,jj
        If(int .eq. 2) then
             k = i
```

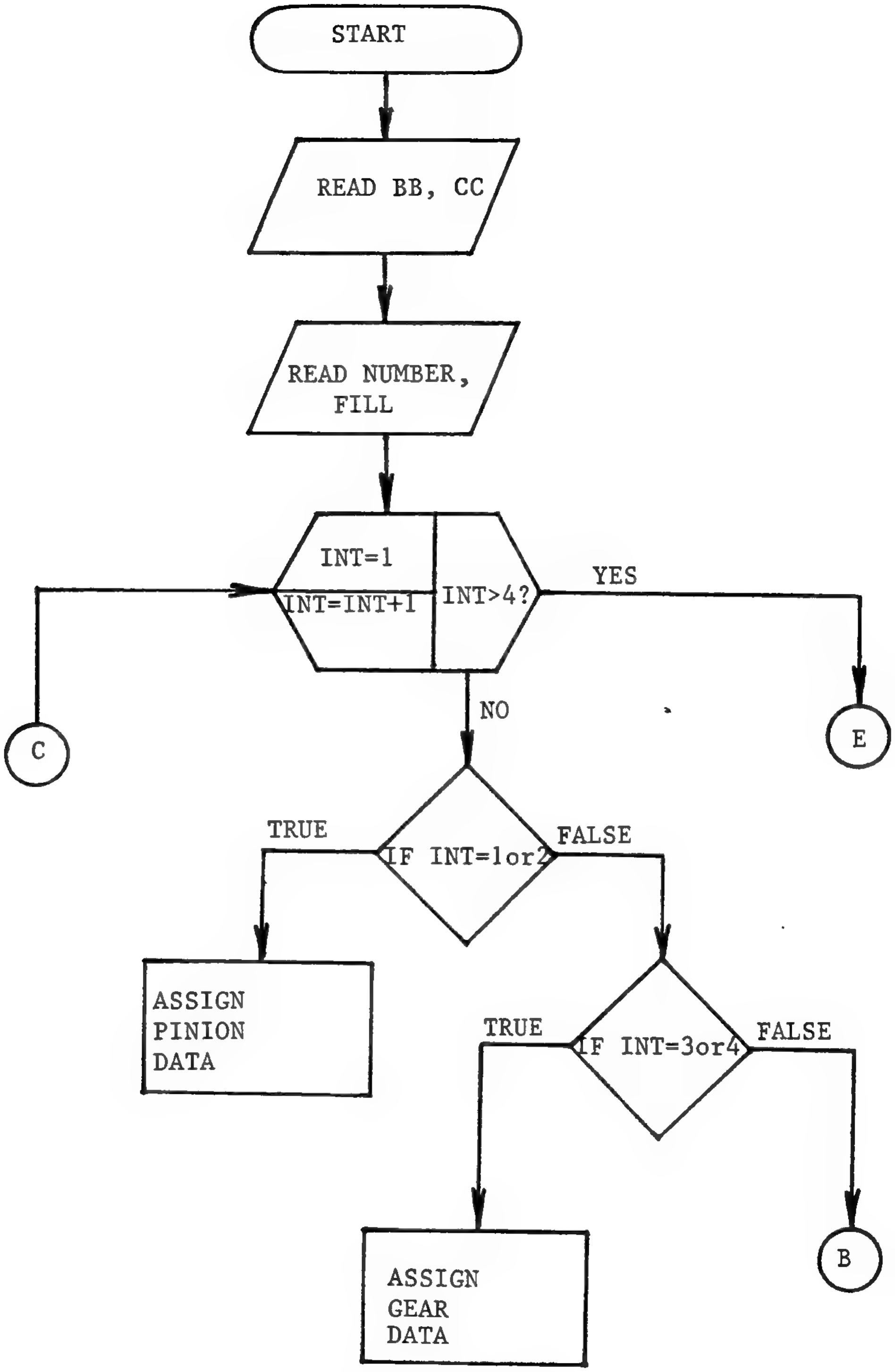
```
            j = k + (cc-1)
       WRITE(nf1,810)'HPAT',',',K,',','2P',',,',K,',',J
       ELSEIF (INT .EQ. 4)  THEN
                K = I + (BB-1)*(cc-1) +3*(bb-1)
                L = (cc-1)*bb + i + 3*bb
                J = L + (cc-1)
       WRITE(NF1,810)'HPAT',',',K,',','2P',',,',L,',',J
       ENDIF
 810   FORMAT(A4,A1,I4,A1,A2,A2,I4,A1,I4)
 800   CONTINUE
C
       RETURN
       END
C
C
C
C
       SUBROUTINE FILLET(XX,YY,ZZ,INT,bb,cc,ri,cl,r1,r2,mu,deden,
     * ntpin,ntge)
C
C      FINDING INNER GEAR BLANK RADIUS GRIDS & ARC LOCATIONS
C
       COMMON/UNITS/NF1,NF2
       INTEGER I,J,K,II,JJ,KK,AR,HP,LI,GR,LS1,LF,PS,PF,PAT
       INTEGER PA,G,NN,HPA,IP,bb,cc,SS
       INTEGER PA1,PA2,PA3,PA4
       INTEGER pa5,pa6,pa7,pa8,pa9,pa10,pa11,pa12
       INTEGER hh1,hh2,hh3,hp1,hp2,hp3,hp4,hp5,hp6,ggb1,ggb2,pppa
       DOUBLE PRECISION PI,RI,MM(100),XX(8000),YY(8000),ZZ(8000)
       DOUBLE PRECISION DELA,DELR,RBAR,R1,R2,ANG,MU,DEDEN,ETA,ROT
       DOUBLE PRECISION PP(100),DELTA(100),X(8000),Y(8000),Z(8000)
       DOUBLE PRECISION XR,YR,ZR,XRHO,YRHO,VEC,ROTAT,cl
       DOUBLE PRECISION RX(100),RY(100),RZ(100)
       INTEGER Li1,Li2,Li3,Li4,Li5,Li6,Li7,Li8,Li9
       INTEGER Li10,Li11,Li12,LI13,Li14,LI15,LI16,Li17,Li18,Li19
       INTEGER W1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14
       INTEGER w15,w16,w17,w18,w19
       INTEGER ntp,ntpin,ntge
C
       LL = 4*(BB*cc)
       DO 125 NN = 1,LL
           X(NN) = XX(NN)
           Y(NN) = YY(NN)
           Z(NN) = ZZ(NN)
 125   CONTINUE
C
C     Do 130 nn = LL +bb, 4*bb*cc +bb
C       x(nn-bb) = xx(nn)
C       Y(nn-bb) = yy(nn)
C       z(nn)    = zz(nn)
C130  CONTINUE
C      FOR INSIDE RADIUS COORDINATES UNDER THE GEAR TOOTH
C
             PI = 3.141592654
       IF(INT.EQ.2)THEN
             GR = LL
             PI = 3.141592654
       ENDIF
C
       IF(INT.EQ.4) THEN
             GR = LL + 6*BB
             PI = 3.141592654
       ENDIF
C
       DO 155 I = 1,BB
       IF(INT.EQ.2)THEN
```

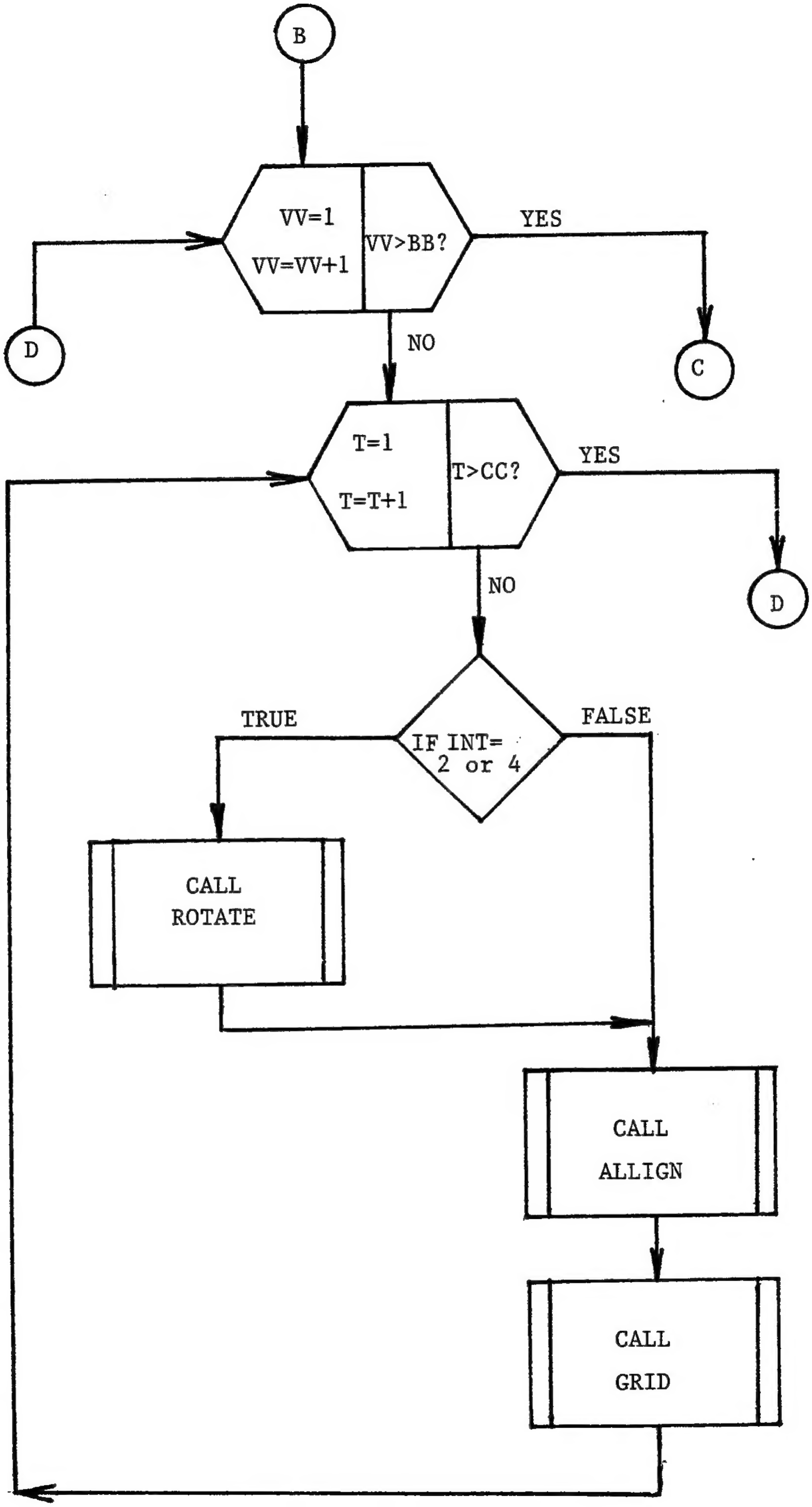
```
        JJ = (I-1)*cc + 1
        KK = (I-1)*cc + BB*cc + 1
        J  = I + BB
        IF (ABS(Y(JJ)).LE.1.0E-5)GOTO 141
        MM(I) = ATAN(Y(JJ)/X(JJ))
        IF (X(JJ).LT.0.0.AND.Y(JJ).LT.0.0)MM(I) = MM(I) + PI
        IF (X(JJ).LT.0.0.AND.Y(JJ).GT.0.0)MM(I) = MM(I) + PI
        GOTO 142
 141    CONTINUE
        MM(I) = 0.0
        IF (Z(JJ).LT.0.0)MM(I) = PI
 142    CONTINUE
        IF (ABS(Y(KK)).LT.1.E-5)GOTO 143
        PP(J) = ATAN(Y(KK)/X(KK))
        IF (X(KK).LT.0.0.AND.Y(KK).LT.0.0)PP(J) = PP(J) + PI
        IF (X(KK).LT.0.0.AND.Y(KK).GT.0.0)PP(J) = PP(J) + PI
        GOTO 140
 143    CONTINUE
        PP(J) = 0.0
        IF (X(KK).LT.0.0)PP(J) = PI
 140    GOTO 155
C
        ELSEIF(INT.EQ.4)THEN
        JJ = (I-1)*cc + 2*(BB*cc)+1
        KK = (I-1)*cc + 3*(BB*cc)+1
        J  = I + 2*(BB*cc) + BB
        IF (ABS(Y(JJ)).LE.1.0E-5)GOTO 151
        MM(I) = ATAN(Y(JJ)/X(JJ))
        IF (X(JJ).LT.0.0.AND.Y(JJ).LT.0.0)MM(I) = MM(I) + PI
        IF (X(JJ).LT.0.0.AND.Y(JJ).GT.0.0)MM(I) = MM(I) + PI
        GOTO 152
 151    CONTINUE
        MM(I) = 0.0
        IF (X(JJ).LT.0.0)MM(I) = PI
 152    CONTINUE
        IF (ABS(Y(KK)).LT.1.E-5)GOTO 153
        PP(J) = ATAN(Y(KK)/X(KK))
        IF (X(KK).LT.0.0.AND.Y(KK).LT.0.0)PP(J) = PP(J) + PI
        IF (X(KK).LT.0.0.AND.Y(KK).GT.0.0)PP(J) = PP(J) + PI
        GOTO 150
 153    CONTINUE
        PP(J) = 0.0
        IF (X(KK).LT.0.0)PP(J) = PI
 150    GOTO 155
        ENDIF
 155    CONTINUE
C
C       FOR INNER GEAR BLANK RADIUS GRIDS & ARC LOCATIONS
C
        IF(INT.EQ.4)THEN
              DELR = (R2-R1)/(BB-1)
              RBAR = R1
        ENDIF
C
        DO 910 I = 1,BB
          J  = I + BB
          G  = GR + 1 + (I-1)*2
C
        IF(INT.EQ.2)THEN
          JJ = (I-1)*cc + 1
          K  = I + 2*BB
          XR = RI*COS(MM(I))
          YR = RI*SIN(MM(I))
          ZR = Z(JJ)
          DELTA(I) = ((PP(J)-MM(I))*180./PI)
          WRITE(nf1,915)’GRID,’,G+bb ,’,,’,XR,’/’,YR,’/’,ZR
          WRITE(nf1,911)'LINE,',k+bb,',ARC,5(0)/1/',delta(i),',',g+bb
  911     FORMAT(a5,i3,a12,f10.5,a1,i4)
C
        ELSEIF(INT.EQ.4)THEN
          JJ     = (I-1)*cc + 2*(BB*cc)+1
          J      = I + 2*(BB*cc)+BB
          K      = I + (cc-1)*BB
          RBAR   = R1 + (I-1)*DELR
          RX(I) = RBAR*COS(MM(I))
          RY(I) = RBAR*SIN(MM(I))
          RICC   = SQRT(X(JJ)*X(JJ)+Y(JJ)*Y(JJ))
          RZ(I) = Z(JJ) + (RICC-RBAR)/TAN(PI/2.-(MU-DEDEN))
          DELTA(I) = (PP(J)-MM(I))*180./PI
          WRITE(nf1,915)'GRID,',G+2*bb,',,',RX(i),'/',Ry(i),'/',Rz(i)
      WRITE(nf1,920)'LINE,',19*bb+i,',ARC,5(0)/1.0/',delta(i),',',g+2*bb
         ENDIF
 915      FORMAT(A5,I4,A2,F10.6,A1,F10.6,A1,F10.6)
 920      FORMAT(A5,I4,A20,F10.4,A2,I4)
 910    CONTINUE
 925    CONTINUE
C
C       TO MAKE HYPAT FROM GEAR BLANK INSIDE RADIUS TO TOOTH BOTTOM
C
        IF(INT.EQ.2)THEN
            GR = 4*BB*cc
            PA = 2*BB*cc - BB
            LI = 3*BB
            HP = 2*(cc-1)*(bb-1) + (BB-1)
            ntp = ntpin
        ELSEIF(INT.EQ.4)THEN
            GR = 4*BB*cc + 6*BB
            PA = 2*BB*cc + BB
            LI = BB*cc
            HP = 2*(cc-1)*(bb-1) +3*(BB-1)
            ntp = ntge
        ENDIF
        NT = FLOAT(NTP)
C
C
           DO 930 I = 1,BB
           LI    = LI + i
         IF(INT.EQ.2)THEN
              Li1 = 4*bb + 2*i
              Li2 = 4*bb*cc +bb + (2*i-1)
              Li3 = 10*bb+(2*i-1)
              Li4 = 10*bb+2*i
              Li5 = Li1
              Li6 = 12*bb+i
              Li7 = 4*bb*cc + 3*bb +(4*i-2)
              Li8 = 4*bb*cc +3*bb +4*i
              LI9 = 13*bb+i
              Li10 = bb*cc+cc*i-(cc-1)
              Li11 = 4*bb*cc+bb+(2*i)
              Li12 = 14*bb+i
              LI13 = 4*bb*cc +3*bb +(4*i-3)
              Li14 = 4*bb*cc +3*bb +(4*i-1)
              LI15 = 15*bb+i
              Li16 = 3*bb+i
              Li17 = 10*bb+(2*i-1)
              II = (I-1)*cc + 1
            DELA = ABS(DELTA(I)*PI/180.)
            ROT  = (2.* PI / NT - DELA)/2.
            VEC  = SQRT(X(II)*X(II)+Y(II)*Y(II))
            ETA  = ACOS(1.-(CL/VEC)**2.)
            XRHO = VEC*COS(MM(I)-ROT)
          YRHO = VEC*SIN(MM(I)-ROT)
          ANG  = ((PI-2.*ROT)*180.)/ PI
          W4 = 8*bb + i
          w5 = 4*bb*cc + i
          W6 = 4*bb*cc + 3*bb + 4*i
     WRITE(nf1,929)'LINE,',4*bb+(2*i-1),',ARC,',xrho,'/',yrho,'/0/',
   *       xrho,'/',yrho,'/1.0/',ang,',',ii
     WRITE(nf1,933)'LINE,',6*bb+(2*i-1),'/',6*bb+2*i,',BR,.5,',
   *        4*bb+(2*i-1)

933  FORMAT (a5,i4,a1,i4,a7,i4)

929  FORMAT(a5,i4,a5,f8.5,a1,f8.5,a3,f8.5,a1,f8.5,a5,f11.5,a1,i4)
934  FORMAT(a5,i4,a3,a2,i4,a1,i4)
       ELSE
              Li1 = 25*bb+(2*i-1)
              Li2 = 4*cc*bb + 8*bb + (2*i - 1)
              Li3 = 27*bb + 2*i
              Li4 = 27*bb + 2*i + 1
              LI5 = 25*bb+(2*i-1)
              Li6 = 29*bb + 1 + i
              Li7 = 4*bb*cc + 10*bb + (4*i -2)
              Li8 = 4*bb*cc +10*bb + 4*i
              Li9 = 30*bb + 1 + i
              Li10 = 3*cc*bb + (cc*i - cc+1)
              Li11 = 4*bb*cc + 8*bb + 2*i
              LI12 = 31*bb + 1 + i
              Li13 = 4*bb*cc + 10*bb + (4*i - 3)
              Li14 = 4*bb*cc + 10*bb + (4*i - 1)
              Li15 = 32*bb + 1 + i
              li16 = 19*bb + i
              Li17 = 27*bb + 2*i
            II = (I-1)*cc + 2*(BB*cc)+1
          DELA = ABS(DELTA(I)*PI/180.)
          ROT  = (2.* PI / NT - DELA)/2.
          VEC  = SQRT(X(II)*X(II)+Y(II)*Y(II))
          XRHO = VEC*COS(MM(I)+ROT)
          YRHO = VEC*SIN(MM(I)+ROT)
          ANG  = -((PI-2.*ROT)*180.)/ PI
      WRITE(NF1,931)'LINE,',20*bb+i,',ARC,','0/0/0/',XRHO,'/',
    *       YRHO,'/',Z(II),'/',ANG,',',II
               w1 = 22*bb + (2*i-1)
               w2 = w1 +1
               w3 = 20*bb+i
               w4 = 24*bb + i
               w5 = 4*bb*cc + 7*bb + i
               w6 = 4*bb*cc + 10*bb + 4*i
      WRITE(nf1,933)'LINE,',w1,'/',w2,',BR,.5,',w3
931   FORMAT(A5,I4,A5,A6,F7.4,A1,F7.4,A1,F10.6,A1,F8.3,A1,I4)
       ENDIF
         IF(INT.EQ.2)ROTAT = -(2 * ROT * 180./PI)
         IF(INT.EQ.4)ROTAT = (2 * ROT * 180./PI)
         IF(I.EQ.1)GR = GR + 1
         IF(I.GT.1)GR = GR + 2
      Write(nf1,932)'LINE,',LI1, ',ARC,0/0/0/0/0/1/',ROTAT,',',LI2
     write(nf1,936)'LINE,',LI3,         '/',LI4,      ',BR,.5,',LI5
     write(nf1,934)'LINE,',w4,',ST',',,',w5,',',w6
    write(nf1,937)'LINE,',LI6,',ST',',,',LI7,',',LI8
     write(nf1,937)'LINE,',LI9,',ST',',,',LI10,',',LI11
     write(nf1,937)'LINE,',LI12,',ST',',,',LI13,',',Lil4
     write(nf1,938)'LI,',Li15,',MER,,',Li16,'/',Li17
938  format(a3,i4,a6,i4,a1,i4)
937   format(a5,i4,a3,a2,i4,a1,i4)
936   format(a5,i4,a1,i4,a7,i4)
932   FORMAT(A5,I4,A18,F8.3,A1,I4)
930      CONTINUE
```

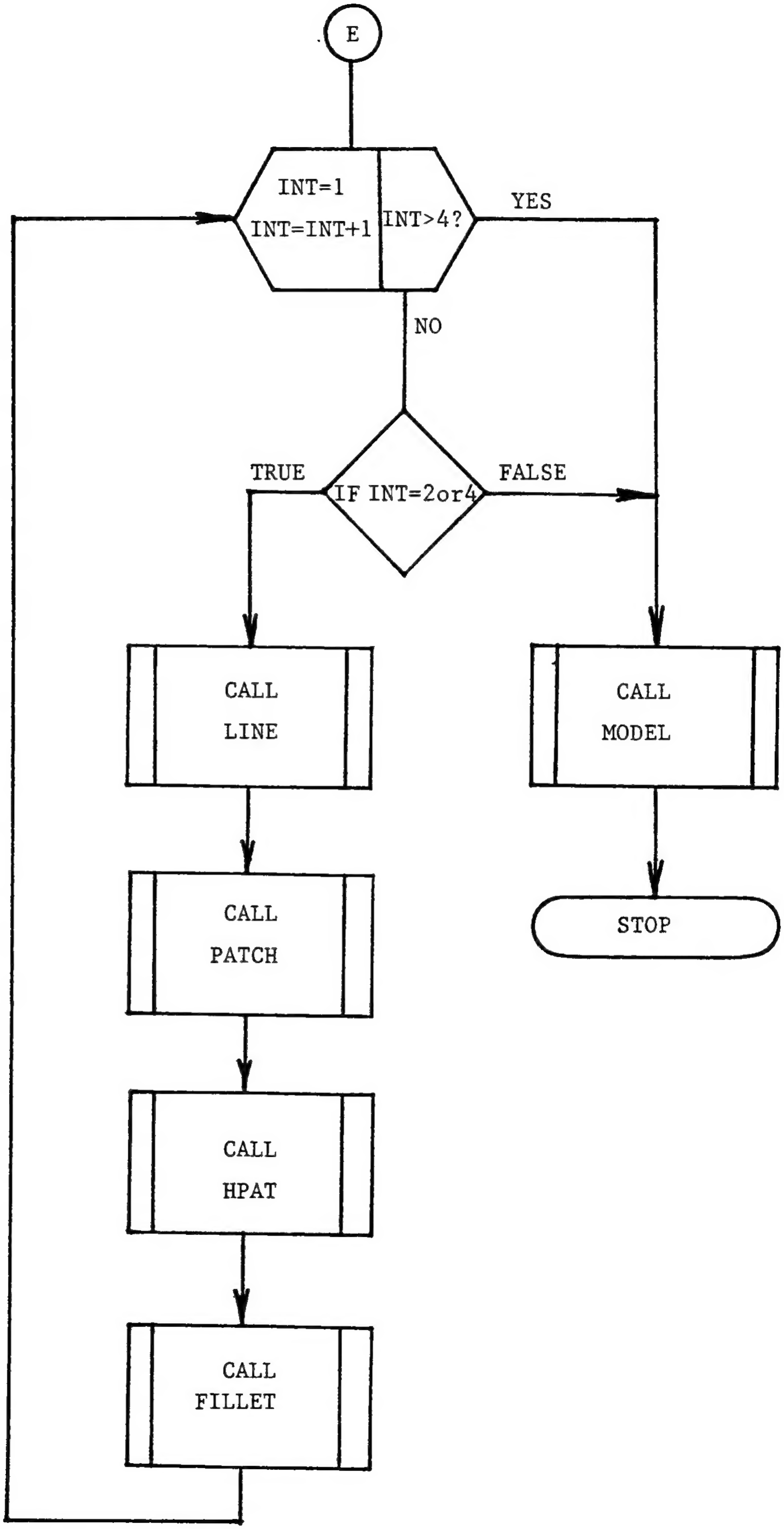
```
C
C        CONNECT LINES ON TOP OF FILLET/ROOT RADIUS &
C               GEAR BLANK INSIDE RADIUS
C
      DO 940 I = 1,bb
       IF(INT.EQ.2) then
        pppa = bb*cc-bb
       pa1 = bb+2*i
       pa2 = 13*bb+i
       pa3 = 15*bb+i
       pa4 = 8*bb+i
       pa5 = bb+(2*i-1)
       pa6 = 8*bb+i
       pa7 = 12*bb+i
       pa8 = 6*bb+(2*i-1)
       pa9 = 10*bb+(2*i)
       pa10 = 14*bb+i
       pa11 = 6*bb+2*i
       pa12 = 12*bb+i
       elseif(int.eq.4)then
       Pppa = 2*bb*(cc-1) + 3*bb
       pa1  = 30*bb + 1 + i
       pa2  = 17*bb + 2*i
       Pa3  = 24*bb + i
       Pa4  = 32*bb + 1 + i
       Pa5  = pa3
       Pa6  = pa2 - 1
       Pa7  = 22*bb + 2*i -1
       pa8  = 29*bb + 1 + i
       pa9  = pa8
       pa10 = pa7 + 1
       Pa11 = 31*bb + i + 1
       pa12 = 27*bb + 2*i + 1
       endif
       write(nf1,941)'PA,',1*PPPA+i,',EDGE,,',pa1,'/',pa2,'/',pa3,'/',pa4
       write(nf1,941)'PA,',1*PPPA+bb+i,',EDGE,,',pa5,'/',pa6,'/',pa7,'/',pa8
       write(nf1,941)'PA,',1*pppa+2*bb+i,',EDGE,,',pa9,'/',pa10,'/',pa11,
     * '/',pa12
  941  format(a3,i4,a7,i4,a1,i4,a1,i4,a1,i4)
 940   CONTINUE
C
C       CONNECT PATCHES IN BETWEEN TEETH TO MAKE HYPERPATCHES
C
       do 945 i = 1,(bb-1)
       If (int.eq.2) then
       hp1 = bb*(cc-1) + i
       hp2 = hp1 + 1
       hp3 = hp1 + bb
       hp4 = hp3 +1
       hp5 = hp3 + bb
       hp6 = hp5 + 1
      hh1 = (bb-1)*(cc-1)+i
      hh2 = hh1 + (bb-1)
      hh3 = hh2 + (bb-1)
      ELSEIF (INT .EQ. 4) THEN
        hp1 = 2*bb*(cc-1) + 3*bb + i
        hp2 = hp1 + 1
        hp3 = hp1 + bb
        hp4 = hp3 + 1
        hp5 = hp3 + bb
        hp6 = hp5 + 1
        hh1 = 2*(bb-1)*(cc-1) + 3*(bb-1) + i
        hh2 = hh1 + bb-1
        hh3 = hh2 + bb-1
        ENDIF
        write(nf1,913)'HPAT,',hh1,',2P,,',hp1,',',hp2
        write(nf1,913)'HPAT,',hh2,',2P,,',hp3,',',hp4
        write(nf1,913)'HPAT,',hh3,',2P,,',hp5,',',hp6
 913   format(a5,i4,a6,i4,a1,i4)
 945   CONTINUE
       IF(INT.EQ.2)GO TO 980
C
 980   CONTINUE
       IF (INT.EQ.2) THEN
           WRITE(NF1,*)'SET,LINES,0'
           WRITE(NF1,*)'NAME,PI 1'
           WRITE(NF1,*)'SET,ACTIVE,NONE'
       ELSEIF(INT.EQ.4) THEN
           WRITE(NF1,*)'SET,LINES,0'
           WRITE(NF1,*)'NAME,GE 1'
           write(nf1,*)'SET,ACTIVE,NONE'
       ENDIF
C
       RETURN
       END
C
C
C
       SUBROUTINE MODEL(BB,cc,number,fill)
C
       COMMON/UNITS/NF1,NF2
C       INTEGER I,J,K,L,numb
       INTEGER I1,I2,J1,J2,K1,K2,L1,L2,bb,cc
C
C             WRITE(NF1,*)'NAME,GE 1,PLOT'
C     to generate the whole gear use this do loop
C       DO 901 I = 1,35
C             L = I + 1
C       WRITE(NF1,905)'NAME,','GE',L,',','RO',',','5(0)/1/10,','GE',I
C 905   FORMAT(A6,A2,I2,A1,A2,A1,A10,A2,I2)
C 901   CONTINUE
C             WRITE(NF1,*)'NAME,GEAR'
            write(nf1,*)'NAME,PI 1,PL'
C           WRITE(NF1,*)'PA,1T#,DEL'
C
            I1 = 1
            I2 = (BB-1)*(cc-1)
        j1 = i2 + 1
        j2 = i2 + 2*(bb-1)
        numb = number/2
C
       write(nf1,917)'MESH,H',i1,'T',i2,',HEX,N,1/',number
    *,'/1/',number,'/1'
          write(nf1,918)'MESH,H',j1,'T',j2,',HEX,N,',numb,'/',
    * fill,'/',numb,'/',fill,'/1'
       write(nf1,919)'MESH,H',j2+1,'T',j2+bb-1,',HEX,N,','3',
    *  '/',numb,'/3/',numb,'/1'
 919  format(a6,i4,a1,i4,a7,a1,a1,i4,a3,i4,a2)
 917  format(a6,i4,a1,i4,a9,i4,a3,i4,a2)
 918  format(a6,i4,a1,i4,a7,i4,a1,i4,a1,i4,a1,i4,a2)
            write(nf1,*)'NAME,PIN'
           WRITE(NF1,*)'SET,ACTIVE,NONE'
           WRITE(NF1,*)'NAME,GE 1,PLOT'
C
            K1 = (bb-1)*(cc-1) + 3  bb-1) + 1
            K2 = K1 + (bb-1) *(cc-1) - 1
            JJ1 = K2 + 1
            jj2 = k2 + 2*(bb-1)
            numb = number/2
C
         write(nf1,917)'MESH,H',K1,'T',K2,',HEX,N,1/',number
    *,'/1/',number,'/1'
```

```
         write(nf1,918)'MESH,H',jj1,'T',jj2,',HEX,N,',fill,'/',
     * numb,'/',fill,'/',numb,'/1'
         write(nf1,921)'MESH,H',jj2+1,'T',jj2+bb-1,',HEX,N,',
     *  numb,'/','3','/',numb,'/','3','/1'


 921      FORMAT(a6,i4,a1,i4,a7,i4,a1,a1,a1,i4,a1,a1,a2)
               K1 = (BB-1)*(cc-1) + 1
               K2 = 2*(BB-1)*(cc-1)
               L1 = 2*(bb-1)*cc + 1
               L2 = 2*(BB-1)*cc + 2*(bb-1)
C     to generate the whole pinion use this do loop
C         DO 902 J = 1,11
C                K = J + 1
C         WRITE(NF1,906)'NAME,','PI',K,',','RO',',','3(0)/1/0/0/30,'
C     *    ,'PI',J
C 906     FORMAT(A6,A2,I2,A1,A2,A1,A14,A2,I2)
C 902     CONTINUE
C               WRITE(NF1,*)'NAME,PINION1'
C
                WRITE(NF1,*)'NAME,GEAR'
                WRITE(NF1,*)'GR,1T#,DEL'
                WRITE(NF1,*)'NAME,PINION,RO,4(0)/1/0/-90,PIN'
                write(nf1,*)'NAME,PINION,PL'
            Return
            END
```

FLOWCHART: PAT. F




B
VV=1
VV=VV+1
VV>BB?
YES
C
D
NO
T=1
T=T+1
T>CC?
YES
D
NO
TRUE
IF INT=
2 or 4
FALSE
CALL
ROTATE
CALL
ALLIGN
CALL
GRID


E
INT=1
INT=INT+1
INT>4?
YES
NO
TRUE
IF INT=2or4
FALSE
CALL
LINE
CALL
MODEL
CALL
PATCH
STOP
CALL
HPAT
CALL
FILLET

FLOWCHART: PAT.F (DISCUSSION)

There are four loops in the main program.

Loop 1: Loops one through four to read data for the four surfaces.

Loop 2: Loops one through BB, where BB is the number of points across the face of the tooth surface.

Loop 3: Loops one through CC, where CC is the number of points along the height of the tooth.

Loop 4: The final loop creates the lines, patches, hyper-patches, fillet and meshes the model for one gear tooth and one pinion tooth.

Subroutines:

1. ROTATE: Rotates the pinion and gear convex sides by Q degrees to create top land.

2. ALLIGN: Aligns the pinion in mesh with the gear.

3. GRID: Creates PATRAN commands for the grids.

4. LINE: Creates PATRAN commands for the lines.

5. PATCH: Creates PATRAN commands for the patches.

6. HPAT: Creates PATRAN commands for the hyper-patches.

7. FILLET: Creates PATRAN commands to creat the fillet.

8. MODEL: Creates PATRAN commands to mesh the model.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 1994

3. REPORT TYPE AND DATES COVERED
Final Contractor Report

4. TITLE AND SUBTITLE

Manual for Automatic Generation of Finite Element Models of
Spiral Bevel Gears in Mesh

5. FUNDING NUMBERS

WU–505–62–36
G–NAG3–1476
1L162211A47A

6. AUTHOR(S)

G.D. Bibel, S. Reddy, and A. Kumar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of North Dakota
Grand Forks, North Dakota 58201

8. PERFORMING ORGANIZATION REPORT NUMBER

E–8671

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Vehicle Propulsion Directorate
U.S. Army Research Laboratory
Cleveland, Ohio 44135–3191
and
NASA Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR–191009
ARL–CR–121

11. SUPPLEMENTARY NOTES

G.D. Bibel, University of North Dakota, Grand Forks, North Dakota (work funded by NASA Grant NAG3–1476); S. Reddy and A. Kumar, University of Akron, Akron, Ohio. Project Manager, Robert F. Handschuh, Army Research Laboratory, Vehicle Propulsion Directorate, organization code 2730, NASA Lewis Research Center, (216) 433–3969.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 37

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The goal of this research is to develop computer programs that generate finite element models suitable for doing 3D contact analysis of faced milled spiral bevel gears in mesh. A pinion tooth and a gear tooth are created and put in mesh. There are two programs: Points.f and Pat.f to perform the analysis. Points.f is based on the equation of meshing for spiral bevel gears. It uses machine tool settings to solve for an N x M mesh of points on the four surfaces, pinion concave and convex, and gear concave and convex. Points.f creates the file POINTS.OUT, an ASCI file containing N x M points for each surface. (N is the number of node points along the length of the tooth, and M is nodes along the height.) Pat.f reads POINTS.OUT and creates the file t1.out. T1.out is a series of PATRAN input commands. In addition to the mesh density on the tooth face, additional user specified variables are the number of finite elements through the thickness, and the number of finite elements along the tooth full fillet. A full fillet is assumed to exist for both the pinion and gear.

14. SUBJECT TERMS

Gears; Gear teeth; Finite elements

15. NUMBER OF PAGES
58

16. PRICE CODE
A04

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT